Journal of the Operational Research Society (2011) 62, 305–312 © 2011 Operational Research Society Ltd. All rights reserved. 0160-5682/11

www.palgrave-journals.com/jors/

Revisiting the big valley search space structure in the TSP

DR Hains*, LD Whitley and AE Howe

Colorado State University, Fort Collins, Colorado, USA

The solution space of the travelling salesman problem under 2-opt moves has been characterized as having a big-valley structure, in which the evaluation of a tour is positively correlated to the distance of the tour from the global optimum. We examine the big-valley hypothesis more closely and show that while the big-valley structure does appear in much of the solution space, it breaks down around local optima that have solutions whose evaluation is very close to that of the global optimum; multiple funnels appear around local optima with evaluations close to the global optimum. The appearance of multiple funnels explains why certain iterated local search heuristics can quickly find high-quality solutions, but fail to consistently find the global optimum. We then investigate a novel search operator, which is demonstrated to have the ability to escape funnels at evaluations close to the global optimum.

Journal of the Operational Research Society (2011) **62,** 305–312. doi:10.1057/jors.2010.116

Published online 1 September 2010

Keywords: combinatorial optimization; local search; travelling salesman problem

Introduction

The neighbourhood of the travelling salesman problem (TSP) under 2-opt moves has been described as having a big-valley structure (Boese *et al*, 1993) in that the evaluation of a tour is positively correlated to the distance of the tour from the global optimal and to other locally optimal tours (see Figure 1). The big-valley structure is conceptualized as a landscape where many local optima may exist, but they are easy to escape and the gradient, when viewed at a coarse level, leads to the global optimum (Freisleben and Merz, 1996; Zhang and Looks, 2005).

However, finding the global optimum has proven to be not so straightforward. Iterated local search techniques such as Chained Lin–Kernighan (Chained-LK) (Applegate *et al*, 2003) are extremely efficient at finding high-quality solutions, but fail to consistently find the global optimal in extended runs for some TSP instances (Applegate *et al*, 2006). To illustrate this point, the implementation of Chained-LK available in the Concorde software package (http://www.tsp.gatech.edu/concorde/) was run 1000 times on the ATT532 instance of the TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/) until no improving tour was found for at least 10 000 iterations. Of these 1000 runs, only four unique tour evaluations were produced. One of these was globally optimal, which was found 55.9% of the time; the other three local optima were each within 0.06% of the global optimum in evaluation.

In this paper, we examine the neighbourhood search structure near the global optimum. Our primary goal is to explain why local search is sometimes trapped in sub-optimal basins from which it cannot escape. We first verify the positive correlation of tour evaluation and distance between local optima of the big-valley structure using random walks from solutions of various evaluations for a number of representative instances.

Having confirmed that the big-valley structure is evident in the central mass of the search space under Lin–Kernighan (LK) search, we hypothesize that the big-valley structure changes when focusing on the area close to the global optimum. To test this hypothesis, we again applied the same random walk procedure to solutions generated by many iterations of Chained-LK. We find that when the walks escape from local optima that have values close to the globally optimal value, the big-valley structure separates into multiple valleys.

The appearance of multiple valleys with large basins of attraction has been called ‘multiple funnels’ in the continuous optimization literature (Doye *et al*, 1999). The multiple funnel concept implies that local optima are organized into clusters, so that a particular local optimum largely belongs to a particular funnel. This means when one escapes from a local optimum, the search almost always moves to another optimum in the same funnel. Thus, the existence of multiple funnels means that once search becomes stuck in one funnel, it is unlikely to escape

*Correspondence: *DR Hains, Department of Computer Science, Colorado State University, 217 Bishop Street, Fort Collins, Colorado 80523, USA.* E-mail: dhains@cs.colostate.edu

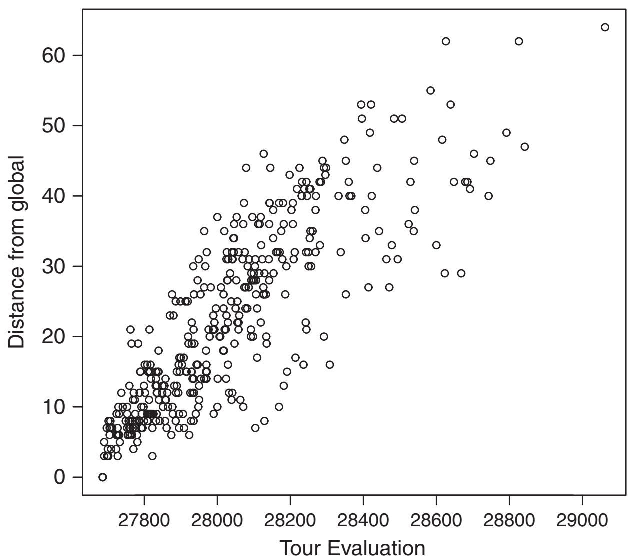

Figure 1 Sampled solutions from LK search using 2-opt on ATT532. The x-axis is the evaluation of the local optimum; y-axis is the distance from the global optimum as measured by the number of shared edges.

and explore a second funnel. Once a local search algorithm gets stuck, a powerful mechanism is needed to drive the algorithm out of the funnel, or restarts are required.

The existence of multiple funnels of low cost tours contradicts the notion of a single 'big-valley'. These funnels were not previously observed because it is difficult to distinguish between local optima in different funnels when optima are sampled across a wide range of evaluations; the multiple funnel structure only becomes clear when focusing on optima with evaluations extremely close to that of the global optimum.

We conjecture that Chained-LK drives the search to the bottom of a funnel and is then highly unlikely to escape to a different funnel. Multiple funnels in the TSP fitness landscape help explain why search algorithms such as Chained-LK do not always find the global optimum even though they consistently find solutions that are near in evaluation to the global.

Some methods may avoid funnels altogether, such as Zhang's backbone search (Zhang and Looks, 2005) or the guided restarts of Lin–Kernighan–Helsgaun (Helsgaun, 2000). However, to escape funnels directly, search needs a different type of operator—one that induces a different connectivity between solutions.

We exploit a new genetic recombination operator, called generalized partition crossover (GPX) (Whitley *et al*, 2009, 2010), to escape funnels. An interesting property of GPX is that if the parents are local optima, the offspring is also almost always local optima. Thus, GPX can move between local optima in a single move. Given local optima from two different funnels, our empirical results indicate that the majority of tours produced by the operator belong to funnels that *differ* from those in which the parents are found.

To further illustrate the ability to escape funnels, we show the performance of GPX in a hybrid genetic algorithm (GA). We find that when controlling for amount of computational effort, the GA using GPX can find better tours on average and continue to improve when compared to Chained-LK.

The random walk method

A random walk method is used to explore the space of local optima near the global. For a given instance of the TSP, two types of initial tours are constructed: one with a high evaluation, v_{high}, and one with a low evaluation, v_{low}. v_{high} is produced by applying steepest descent 2-opt to a randomly generated tour. v_{low} is produced by applying 200 iterations of Chained-LK search that takes a random double bridge move in which four edges are replaced at random after each iteration of LK search (Lin and Kernighan, 1973; Johnson, 1990; Johnson and McGeoch, 1997). We use the implementation of LK search from the Concorde software package, which uses do not look bits and candidate lists.

In our experiments, we rely on random walks to escape from one basin of attraction into another. We start the walk at a locally optimal tour v_0, which is initialized to v_{low} or v_{high}. At step i of the random walk, a new tour v_i is produced by applying a random 2-opt move to v_{i-1}, by breaking two random edges in v_{i-1}, $e(a, b)$ and $e(c, d)$, and replacing them with $e(a, c)$ and $e(b, d)$ to form the tour v_i. After each step, an LK search is applied to v_i until a locally optimal tour v^* is found. If v^* differs from v_0, v_i has escaped the basin of attraction surrounding v_0; i is said to be the escape length and the random walk is halted.

Our first experiment to confirm the big-valley hypothesis was run on three test instances. Two of these instances, ATT532 and PCB442, are from TSPLIB. ATT532 is 532 cities from the continental United States. PCB442 is a highly structured representation of a printed circuit board. The third instance, denoted RAND532, is a uniformly random Euclidean instance created by placing 532 points in a grid of size 1 000 000 by 1 000 000.

Additional problems are also used to test hypotheses later in this paper. They are NRW1379, U574, U1817, PCB3038, RAND500, RAND1500 and RAND3000. Together with the other instances, these comprise a set of four random, four highly structured and two city problems of varying sizes.

Confirming the correlation of the big valley

We first show that ATT532, PCB442 and RAND532 exhibit the correlation of distance and evaluation of the big-valley structure using random walks. Let the function C denote the evaluation function. The evaluation of each of the initial tours for ATT532 is $C(v_{high}) = 30\,947$ and $C(v_{low}) = 27\,710$, for PCB442 $C(v_{high}) = 55\,058$ and $C(v_{low}) = 50\,823$, and for RAND532 $C(v_{high}) = 18\,778\,884$ and $C(v_{low}) = 16\,903\,785$.

For each of the test instances, a set of 500 local optima denoted by L was generated by doing random walks to escape from v_{low}, and a set of 500 optima denoted by H was generated by doing random walks to escape from v_{high}. A 'bond distance' $d(v_i, v_j)$ defined in Boese *et al*, 1993, 1994 is computed for each local optima v^* found on a random walk from the initial tour v_0 by LK search. Let N denote the instance size, and let $s_{i,j}$ denote the number of shared edges between tours v_i and v_j, so that:

$$d(v_i, v_j) = N - s_{i,j}$$

The values d_{low} and d_{high} in Table 1 are the average bond distances of the local optima v^* found from the walks starting at v_{low} and v_{high}.

The P-values reported in Table 1 are from a t-test using the alternative hypothesis $d_{low} - d_{high} < 0$. Under the big-valley hypothesis, we expect this to be true as the distance between lower cost tours is expected to be less than those of higher cost tours.

The Pearson correlation coefficient r was calculated on the distance *versus* evaluation for each of the 1000 local optima (500 from each of the two starting tours). The almost perfect linear correlation produced by LK search further strengthens the big-valley hypothesis as it shows that solutions of low evaluations tend to cluster closer to one another than do tours of high evaluations under LK search.

However, as we will show in a later section, these clusters of low-cost solutions are not all concentrated around the global optimum forming a single big-valley. Instead they form distinct pockets or funnels in the search space that are located a non-trivial bond distance from one another.

Table 1 Mean distance (d_{high}, d_{low}) and evaluation distance (δ_{high}, δ_{low}) from the initial local optima v_{high} and v_{low} and the local optima found by LK search along random 2-opt walks from the initial optima

Instance	*ATT532*	*PCB442*	*RAND532*
d_{high}	176.7 ± 0.71	125.4 ± 0.85	155.9 ± 0.80
d_{low}	17.5 ± 1.6	19.7 ± 1.3	13.3 ± 1.2
p-value	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
δ_{high}	-2809 ± 8	-3335 ± 14	$-160\,355 \pm 4341$
δ_{low}	125 ± 11	222 ± 8	$43\,549 \pm 3387$
r	0.99	0.98	0.99

The Pearson correlation coefficient (r) between distance and evaluation of the local optima found by LK search and the p-value of a t-test with alternative hypothesis $d_{low} - d_{high} < 0$ are also reported.

The variables δ_{low} and δ_{high} are the average differences in evaluation between new found optima and the initial tours. A positive δ value indicates that poorer quality tours than that of the initial tour were found on average. Interestingly, the local optima found after escaping v_{high} tended to improve over the starting local optimum, while the optima surrounding v_{low} tend to be worse. This suggests that even though the v_{low} tours are not the global optima, the majority of surrounding optima are of a lower quality. This could make it difficult for an iterated local search that accepts only improving moves to escape v_{low} (or an optimum with similar evaluation) if the perturbation operator can only move a short distance in the search space.

Structure close to global optima in the TSP

We conjecture that the big-valley structure breaks down as search approaches the global optimum, making it harder to discover the global optimum. The clustering of low-cost solutions still takes place, but these clusters tend to form pockets or funnels that contain several solutions who evaluations are within a small percentage of the global optimum. In addition, the bond distance between these funnel bottoms is non-trivial, making it nearly impossible for Chained-LK to escape the funnel.

We took two approaches to assessing the structure of the search space near the global optimum. First, we examined the number of unique tours found after considerable search effort. We ran Chained-LK until at least 10 000 iterations are performed without finding an improving tour. Chained-LK was run in this manner for 1000 runs, recording the best tour found from each run. Of these 1000 tours, we only keep unique tours, that is, a tour must have at least one different edge from all other tours produced. We collected data on 10 instances; Table 2 shows relatively few unique evaluations across the 1000 runs.

Second, we explored the regions around the unique evaluations. For this analysis, we focus on ATT532 because it has just four unique evaluations which can be easily displayed graphically. For ATT532, each set of unique tours with the same evaluation formed a connected component in the search space under 2-opt; so the tours are considered to be on plateaus. One tour from each of the four plateaus was selected randomly to represent the funnel bottom (the largest plateau was of size six). Five hundred local optima were found with the random walk procedure starting from each of the four funnel bottoms. Figure 2 plots the distance of the tours to the global optimum[1] *versus* the tour evaluations; the majority of local optima are further away from the global than the tours associated with the bottoms of each funnel.

[1] In fact, we found two distinct globally optimal solutions. Because they were close together, we selected one for distance computation.

Table 2 Number of unique tours and evaluations found after running Chained-LK for 1000 runs on 10 instances

Instance	*Number of unique tours*	*Number of unique evaluations*
Random Euclidean instances		
RAND500	3	2
RAND532	1	1
RAND1500	194	151
RAND3000	979	432
Somewhat structured instances (city problems)		
ATT532	20	4
NRW1379	902	71
Highly structured instances (PCB, drilling problems)		
PCB442	1	1
U574	13	4
U1817	967	264
PCB3038	980	305

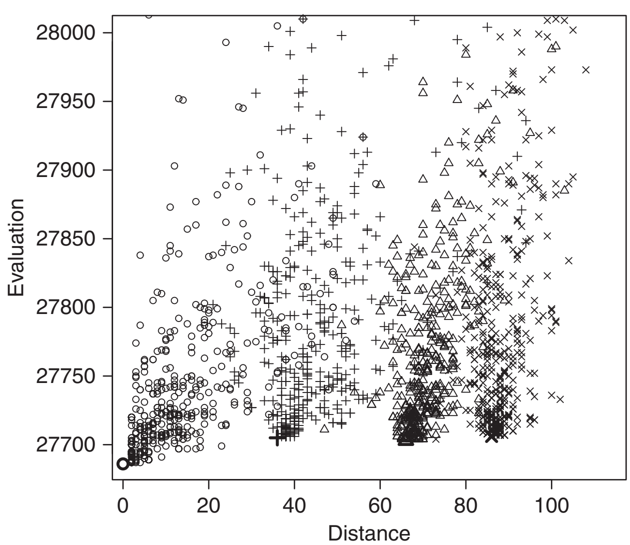

Figure 2 Five hundred local optima found by LK search when starting at each of the four funnel bottoms in ATT532. The initial tour of the walk is shown with the symbol matching its evaluation shown in the legend; locally optimal tours are shown with a slightly smaller symbol corresponding to the symbol of the initial tour of the walk.

These plots concentrate attention on part of the search space very close to the global; $C_{high} = 30\,947$ does not appear on any of these plots. The local optima found in each funnel are correlated in evaluation and distance to the funnel bottoms and other tours in that funnel coinciding with the big-valley hypothesis. However, a correlation does not exist between evaluation and distance to the global optimal.

Each funnel itself is a non-trivial bond distance from the other funnels though this is not evident in Figure 2.

Table 3 Pairwise distance of each funnel bottom used as an initial tour in Figure 2

Evaluation	27 686	27 703	27 705	27 706
27 686	0	66	36	86
27 703	66	0	76	43
27 705	36	76	0	73
27 706	86	43	73	0

For example, the funnels with bottoms corresponding to plateaus with evaluations of 27 705 and 27 706 seem to be close to one another in Figure 2. However, this is only because we are measuring the distance relative to the global optima, the bond distance between the two tours is actually 73. The pairwise distances between the funnel bottoms are in Table 3.

Although we only report detailed results for ATT532, similar results were found in all the instances tested. Table 2 shows that the number of funnels tends to grow with the instance size for all problems. Table 3 shows that the funnels corresponding to tours of unique evaluations were always separated by a non-trivial distance for instance ATT532. The pairwise distances between funnels also increased with the instance size for all instances.

Rather than a single big-valley, there are multiple funnels associated with local optima close in evaluation to the global optimum. For LK search, the big-valley structure holds within funnels but not across funnels below a certain evaluation.

Implications for chained-LK performance

We now return to the conundrum of why local search is not more effective at finding the global optimum in TSP. This answer is that the perturbation performed by the double bridge move is too small. Consider the distances in Table 3. The lowest distance between the global optimum and a non-globally optimal solution is 36 for ATT532 (this distance only grows with instances of larger size). The average distance of local optima found by LK-search when taking a random walk is 17.5 ± 1.6 from optima of low evaluation (Table 1). The average length of the random walks is 2.10 ± 0.04, which corresponds to two random 2-opt moves. It is highly unlikely that LK search can hop from a non-globally optimal funnel bottom to the globally optimal solution by perturbing a low cost tour using random double bridge moves.

The random double bridge move used by Chained-LK is roughly equivalent to two random 2-opt moves (four edges go out and four edges go in). It does, however, sample a different space than two random 2-opt moves would. If either of the two pairs of edges, which are replaced by the double bridge move, were broken and

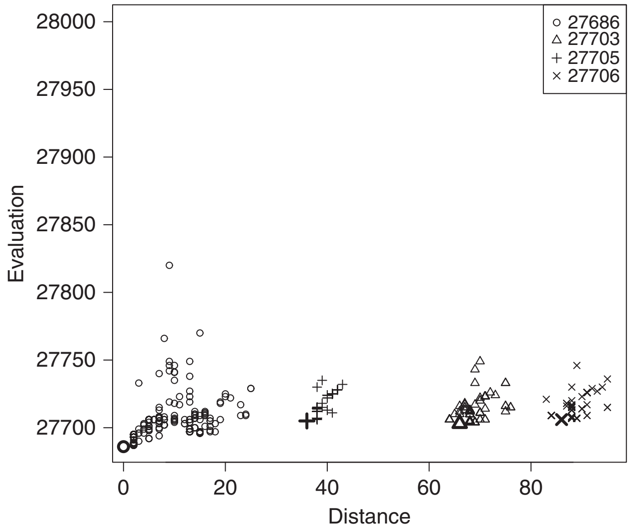

Figure 3 Five hundred local optima found by LK search after perturbing the four funnel bottoms from ATT532 by double bridge moves.

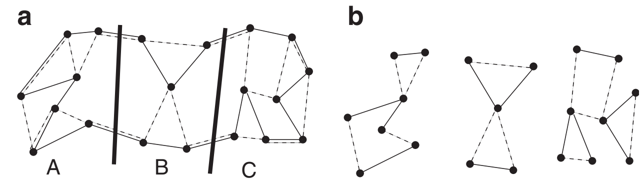

Figure 4 An example of the union graph and connected components created by the PX operator when recombining two tours. The edges from one tour are represented by the solid line, the other by the dashed line. On the left (a) the union of two tours with the partitions of cost two shown as heavy lines. On the right (b) are the connected components of the union graph after common edges are removed.

replaced individually, a cycle would be formed, making the resulting solution invalid. Therefore, the random double bridge move consists of two *invalid* random 2-opt moves, moves that would not be performed by our random 2-opt walk.

To extend our observations to double bridge moves, we repeated the random walk experiment from funnel bottoms, only using random double bridge moves instead of 2-opt to perturb the tour until a new local optima was found. The results of this experiment are shown in Figure 3. Of the 500 tours found from each initial tour, almost all of them are the same and are clustered even closer to the original tours than in Figure 2. The double bridge operator actually exacerbates the funnels. Both random double bridge moves and random 2-opt walks do not move far enough in the space to escape a funnel.

Escaping funnels with GPX

In the light of our search space analysis, we will discuss a recombination operator that can locate new funnels given two parent tours from different funnels. GPX takes two solutions and recombines them so that (1) all common edges in the two solutions are inherited and (2) all edges that are passed to offspring must be present in the parent solutions. GPX works by constructing a graph G from the union of two tours and finding all partitions of cost two on that graph. Edges that are common to both the parents tours are considered to be a single edge in graph G. Therefore, a partition of cost two occurs if and only if the partition cuts two edges that both appear in both parents. The GPX operator as implemented is guaranteed to be feasible if and only if there is a partition of cost two in graph G.

Figure 4a shows an example of the union of two tours with two partitions of cost two shown by heavy solid lines. GPX finds these partitions in $O(n)$ time by removing the common edges found in both tours from G (Figure 4b) and performing a breadth first search on the reduced graph. Each vertex in G is visited only once, and no vertex has degree greater than 4. Label one of the parents used to construct graph G as the 'solid' tour and the other as the 'dashed' tours as illustrated in Figure 4a. By cutting graph G at every partition of cost two, the graph is divided in c subgraphs, which we will denote as *components*. If there are c components, we can locate 2^c Hamiltonian circuits in graph G, all of which include the common edges in G. However two of these tours are the two 'parents' used to construct G. To perform recombination, an 'offspring' inherits all of the common edges in G. Within each *component* of G, the offspring inherits edges only from one parent, either dashed or solid edges. Of the 2^c Hamiltonian circuits in graph G, the best of all of these Hamiltonian circuits can be generated by greedily picking the shortest path in each component.

We have previously proven that this process always results in a Hamiltonian circuit (Whitley *et al*, 2009, in submission). To see why this is true, consider a single partition. Cutting the graph G once creates two components; since the partition cost is exact two, each component has a single entry and a single exit. Label the two components the 'left hand side' (LHS) and the 'right hand side' (RHS). The construction of a new offspring follows a path that enters the LHS component, follows either the dashed tour or the solid tour and then exits the component. When the RHS component is entered, the decision to follow the dashed tour or the solid tour is independent of what happened in the LHS component.

In Figure 4a, one can observe that not all components have exactly one entry and one exit. However, since every offspring inherits all common edges, they also inherit every partition found in the parents. Also, note that if recombination occurs at only one partition, only two offsprings are possible. Furthermore, if one merges the two offspring, or if one merges the two parents, one obtains the exact same graph G. Thus, generalized partition crossover over c components can be viewed as multiple interative applications of recombination utilizing one partition at a time in graph G.

When two offspring are produced under GPX, there are 2^c-2 possible offspring. In the remainder of this paper, one of these offspring will be the 'greedy' in construction, that is, GPX picks the shortest path in each component. The other offspring will be generated by being 'greedy' in every component except for the largest component. In the largest component the alternate, longer path will be selected to create the second offspring.

Assessing the ability of GPX to escape funnels

We show empirically that the GPX operator escapes funnels. To do so, we choose two distinct funnel bottoms from each instance used in the previous experiment. Only funnels that were not associated with the global optimum were chosen. To generate parent tours associated with a funnel, the random walk method starting from the funnel bottoms was used to find 50 local optima within each funnel. When Chained-LK is applied to these local optima, the search always ends back in the funnel bottom from which the random walk was initiated.

To test the effectiveness of the GPX operator in escaping funnels, we apply the operator to all pairs of tours such that each pair consists of one tour from each of the two funnels. This results in 2500 applications of the GPX operator per instance. Two offsprings are generated, one of which is the result of greedy recombination. Chained-LK is then applied to the two tours that are produced by GPX until it finds no improving tour for at least 10 000 iterations. If one of the tours found by Chained-LK is in a different funnel from the initial funnels, we conclude that a single application of the GPX operator was successfully able to escape the initial funnels.

The results are presented in Table 4. RAND532 and PCB448 were not tested because they were found to have only one funnel (see Table 2). In all other instances, the GPX operator displayed the ability to produce a tour that led to a different funnel than the funnel in which the parents were found.

Table 4 Number of funnel escapes out of 2500 crossovers when applying GPX to two tours in different funnels

Instance	RAND500	RAND 1500	RAND3000	ATT532
Escapes	1288 (51%)	1655 (66%)	639 (26%)	1129 (45%)
Instance	NRW1379	U574	U1817	PCB3038
Escapes	2254 (91%)	940 (38%)	1502 (60%)	1474 (58%)

The hybrid GA

To determine if operator effectiveness translates to search results, we compared a hybrid GA using GPX and LK search against Chained-LK using double bridge moves. At each generation, the hybrid GA recombines the best tour in a population of 10 tours with the remaining nine tours in the population and then a single application of LK search is applied to the most diverse 10 tours of the offspring (Whitley *et al*, 2010). The hybrid GA is described in Figure 5. The initial population is produced by randomly generating t tours and applying the same LK search procedure used in Step 5. The version of LK search used is exactly the same as the LK search routine used by Chained-LK (Applegate *et al*, 2003) with do not look bits and using the default neighbourhood list size and search depth. The hybrid GA is executed for a fixed number of generations.

One of the limitations of GPX is that it can only recombine edges that are currently in the population. We developed a strategy called *diversity selection* that uses an edge weighting function d to quantify the diversity of edges contributed to the population by each tour. For tour s_i in the population,

$$d(s_i) = \sum_{e(j,k)\in s_i} \frac{1}{M(j,k)}$$

where $e(j,k)$ is an edge from city j to k and $M(j,k)$ is the number of times $e(j,k)$ appears in the population. We then retain tours from among the offspring with the highest summed edge diversity, $d(s_i)$. The use of diversity selection means that the GA must be generational and that offspring replace parents because parents typically have higher diversity than offspring.

At every generation, we retain the *best* tour found so far (Step 3) in the population of offspring. If GPX fails to recombine two tours, we apply one double-bridge move to tour i, where i is *not* the best tour in the population, and directly place this 'mutated' tour in the population of offspring (Step 2). The remaining members of the offspring population are selected by diversity selection (Step 4).

Let $P1$ be a randomly generated population of size t;
Let $P2$ be a temporary child population of size t;
For each member of $P1$: apply LK-search and evaluate;

1. Attempt to recombine the best tour of $P1$ with the remaining $t-1$ tours using GPX; this generates a set of up to $2t$ offspring.
2. If recombine was not feasible between the best tour and tour i, mutate tour i using a double bridge move and place in population $P2$;
3. Place the best solution found so far in population $P2$;
4. From the set of offspring, select offspring to fill population $P2$;
5. For each member of population $P2$: apply LK-search and evaluate;
6. $P1 = P2$; If stopping condition not met, goto 1;

Figure 5 Algorithm for the hybrid GA; the GA is generational, but elitist.

Comparing to chained-LK

To compare effectiveness, we allowed Chained LK and the hybrid GA to call the LK search subroutine exactly the same number of times. Since the population size of the hybrid GA is 10, this means that Chained-LK is allowed to call LK search 10 times for every generation that the hybrid GA is allowed to execute. Chained-LK uses a double bridge move after every call to the LK search subroutine. The running times of the double bridge moves and GPX are insignificant compared to the $O(n^2)$ running time of LK search. Also, the double bridge move is always a disimproving move, while the GPX recombination is almost always an improving move (and often a local optimum). Thus, a call to the LK search subroutine is usually more expensive after a double bridge move compared to after recombination. Both Chained-LK and the GA use the same implementation of LK search and are allowed to call it 50 000 times before terminating.

The average cost above optimal of the best tour found in over 50 trials on several instances is shown in Table 5 at several points during the computation. As the search progresses, Chained-LK tends to stagnate and find no more improving moves as the tours continue to settle back into funnel bottoms. The hybrid GA is capable of continually improving upon the minimum tour found with more search calls and even finds the optimal instance in every trial for U574 and RAND500.

Figure 6 depicts the same cost *versus* distance plot as in Figure 3 except that the tours are found by the hybrid GA. It does not show the clustering that exists in Figures 3 and 1 and displays a more even distribution of tours over the lower part of the search space. These results were obtained after 10 010 calls to the LK search subroutine. Our experiments suggest that this distribution of tours can be leveraged to find the globally optimal solution more frequently.

Conclusions

Our experiments have shown that while the search space overall appears to exhibit the big-valley structure, the space consisting of tours with evaluations near that of the global optimum does not. The existence of multiple funnels may explain why some local search techniques can generate near optimal solutions, and yet are sometimes unable to find a global optimum even when search is extended for a

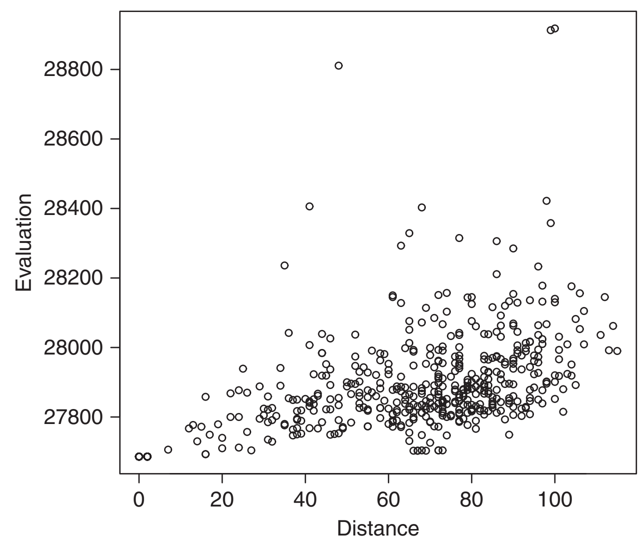

Figure 6 Distance to optimal (x-axis) *versus* cost function evaluation (y-axis) of the 500 tours produced by the GA using GPX on ATT532 after 10 010 LK search call.

Table 5 The average percentage above optimal for the GA using GPX and Chained-LK using double bridge moves

Instance	*LK search calls*				
	10 010	20 010	30 010	40 010	50 010
rand500 (GA w/GPX)	0±0	0±0	0±0	0±0	0±0
rand500 (Chained-LK)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
att532 (GA w/GPX)	0.03±0	0.02±0	0.02±0	0.01±0	0.01±0
att532 (Chained-LK)	0.03±0	0.03±0	0.03±0	0.03±0	0.03±0
u574 (GA w/GPX)	0.01±0	0±0	0±0	0±0	0±0
u574 (Chained-LK)	0.05±0.01	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01
nrw1379 (GA w/GPX)	0.06±0	0.06±0	0.05±0	0.04±0	0.04±0
nrw1379 (Chained-LK)	0.06±0.01	0.06±0	0.05±0	0.05±0	0.05±0
rand1500 (GA w/GPX)	0.05±0.01	0.04±0.01	0.03±0.01	0.03±0.01	0.02±0.01
rand1500 (Chained-LK)	0.09±0.01	0.08±0.01	0.08±0.01	0.08±0.01	0.08±0.01
u1817 (GA w/GPX)	0.23±0.02	0.19±0.01	0.18±0.01	0.17±0.01	0.15±0.01
u1817 (Chained-LK)	0.39±0.02	0.35±0.02	0.33±0.02	0.32±0.02	0.31±0.02

The results are averaged over 50 trials. Both algorithms were given the same number of LK search calls that are indicated in the table.

prolonged amount of time. Once the search is driven to the locally best tour in one of these funnels, it can be difficult if not impossible for a search algorithm that uses perturbation methods to escape the funnel. This observation motivates further research into strategies that can avoid or escape these funnels.

The GPX operator is capable of escaping funnels when the double bridge move cannot. The operator has been shown empirically to be able to escape funnels when recombining solutions from different funnels. When used in conjunction with LK search in a hybrid GA, GPX can continually improve tours with subsequent LK search calls. Chained-LK eventually gets stuck and is unable to improve on the best so far tour even with a large number of additional calls to LK search. The hybrid GA does not get stuck as the ability of GPX to escape funnels allows it to further improve the best so far tours after the point at which Chained-LK gets stuck.

Acknowledgements—This research was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA9550-08-1-0422. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

Applegate D, Bixby R, Chvatal V and Cook W (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press: Princeton, NJ.

Applegate D, Cook W and Rohe A (2003). Chained Lin–Kernighan for large traveling salesman problems. *INFORMS J Comput* **15**(1): 82–92.

Boese KD, Kahng AB and Muddu S (1993). *On the big valley and adaptive multi-start for discrete global optimizations*. Technical report, UCLA CS Department, TR-930015.

Boese KD, Kahng AB and Muddu S (1994). A new adaptive multi-start technique for combinatorial global optimizations. *Opns Res Lett* **16**: 101–113.

Doye JPK, Miller MA and Wales DJ (1999). The double-funnel energy landscape of the 38-atom Lennard-Jones cluster. *J Chem Phys* **110**: 6896.

Freisleben B and Merz P (1996). A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. *Proceedings of IEEE International Conference on Evolutionary Computation,* pp 616–621, doi:10.1109/ICEC.1996.542671.

Helsgaun K (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *Eur J Opl Res* **126**: 106–130.

Johnson DS (1990). Local optimization and the traveling salesman problem. *Lect Notes Comput Sc* **443**: 446–461.

Johnson DS and McGeoch LA (1997). The traveling salesman problem: A case study in local optimization. *Local Search Comb Optim*: 215–310.

Lin S and Kernighan BW (1973). An effective heuristic algorithm for the traveling salesman problem. *Opns Res* **21**: 498–516.

Whitley D, Hains D and Howe A (2009). Tunneling between optima: Partition crossover for the traveling salesman problem. In: Raidl *et al* (eds) *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM: NY, pp 915–922.

Whitley D, Hains D and Howe A (2010). A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover. In: Rudolph G *et al* (eds). *Proceedings of 11th International Conference on Parallel Problem Solving From Nature (PPSN 2010)*. Springer: Berlin.

Zhang W and Looks M (2005). A novel local search algorithm for the traveling salesman problem that exploits backbones. *Int Joint Conf Artif Int* **19**: 343.

Received June 2009;
accepted May 2010 after one revision