

# A Hybrid Genetic Algorithm for the Traveling Salesman Problem using Generalized Partition Crossover

Darrell Whitley, Doug Hains, Adele Howe

Colorado State University, Fort Collins, CO 80524

**Abstract.** We present a hybrid Genetic Algorithm that incorporates the Generalized Partition Crossover (GPX) operator to produce an algorithm that is competitive with the state of the art for the Traveling Salesman Problem (TSP). GPX is respectful, transmits alleles and is capable of tunneling directly to new local optima. Our results show that the Hybrid Genetic Algorithm quickly finds optimal and near optimal solution on problems ranging from 500 to 1817 cities using a population size of 10. It is also superior to Chained-LK given similar computational effort. Additional analysis shows that all the edges found in the globally optimal solution are present in a population after only a few generations in almost every run. Furthermore the number of unique edges in the population is also less than twice the problem size.

**Key words:** Traveling Salesman Problem, Generalized Partition Crossover, Hybrid Genetic Algorithm, Chained-LK

## 1 Introduction

The state of the art for the Traveling Salesman Problem (TSP) is Chained Lin-Kernighan (Chained LK) [6], a local search algorithm coupled with a carefully designed operator called the “double bridge move”. In this paper, we present an evolutionary algorithm that is competitive with Chained LK and that can be shown to possess several desirable characteristics. Our hybrid Genetic Algorithm combines local search with a new recombination operator, Generalized Partition Crossover (GPX), which is a generalization of our Partition Crossover operator (PX) [2].

PX is respectful and transmits alleles: this means that the children of parent solutions are guaranteed to have all edges that are found in both parents (respect) and any edge found in an offspring can also be found in one of the parents

---

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

(transmits alleles) [1]. Additionally the operator has a property which we call “tunneling”: the offspring of parents that are locally optimal are also locally optimal with high probability.

Partition Crossover (PX) partitions a graph  $G$  constructed from the union of two parent tours. If a partition of cost 2 exists in this graph, PX is able to construct two children which are distinct from the parents in  $O(n)$  time, where  $n$  is the number of vertices in graph  $G$ . If there are multiple ways to partition graph  $G$  into subgraphs, where each partition has cost 2, PX uses only one of these partitions. But empirically we have found that multiple partitions exist when recombining solutions that are already locally optimal. GPX exploits *all* partitions of cost 2 with no significant increase to the  $O(n)$  running time of the original PX operator; the resulting recombination is still respectful and transmits alleles.

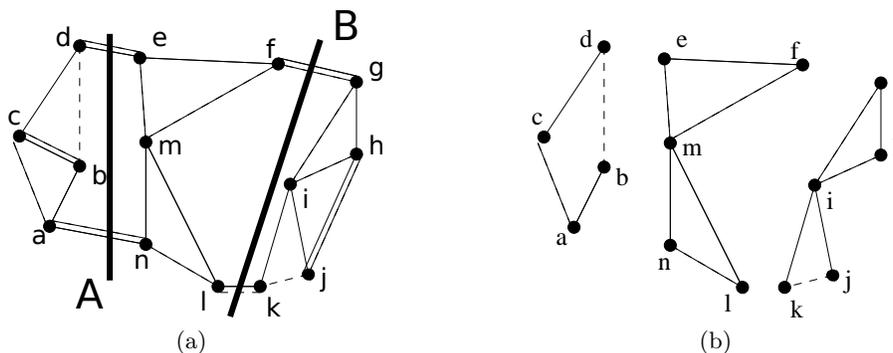
When we embed GPX in a hybrid GA, we can show improvements in efficiency and effectiveness over Chained LK. We demonstrate its performance in experiments in which we carefully control computational effort to provide a fair comparison. On small 500 city problems the Hybrid Genetic Algorithm often quickly finds the global optimum. It occasionally finds the global optimum on larger problems instances using a very modest amount of computational effort as compared to Chained LK.

More importantly, we analyze the cases when the hybrid GA does *not* easily find the global optimum. We find that the edges in the global optimum which may be missing from the best tour are present in other members of the population. Furthermore, the number of unique edges in the population is relatively small compared to the total number of edges in the cost matrix. In addition, the edges from the global optimum which are missing from the best solutions are predominantly contained within a single subgraph which is the largest “component” of the graph that is being broken apart during the recombination.

## 2 Generalized Partition Crossover

To recombine two Hamiltonian circuits, Partition Crossover partitions a graph  $G = (V, E)$  where  $V$  is the set of vertices (i.e., cities) of an instance of a TSP and  $E$  is the union of the edges found in two parents. An edge in  $E$  can be classified as either a *common* edge or an *uncommon* edge. An edge in  $E$  is a *common* edge if it is found in both parents; an edge is an *uncommon* edge if it is found in only one parent.

Whitley et al. [2] prove that if graph  $G$  contains at least one partition of cost 2, then it is possible to create at least two offspring in  $O(n)$  time which are Hamiltonian circuits distinct from the parents. Figure 1(a) shows a graph  $G$  created from two parents. The edges from one parent are represented by solid lines and those from the other by dashed lines. When the common edges are deleted, the graph breaks into 3 subgraphs. There are two partitions of this graph with cost 2 (the heavy dark lines).



**Fig. 1.** An example of (a) The graph  $G$  created from the union of two parent tours with two partitions shown by the heavy dark lines and (b) The graph  $G_u$ , constructed by deleting the common edges between the two parent tours from  $G$

The original PX operator constructs two children using only *one* of the partitions of cost 2 in  $G$ . PX could construct two children using partition A in Figure 1 by taking the solid edges from the left of A and the dashed edges from the right, and the second child by taking the dashed edges from the left of A and the solid edges from the right. Two different children would be constructed if PX used partition B in a similar manner. Whitley, Hains and Howe [2] prove that Partition Crossover is respectful and transmits alleles.

Generalized partition crossover (GPX), makes use of all partitions of cost 2 in a single recombination with no significant increase to the  $O(n)$  running time of PX. We recombine solutions by creating a subgraph of  $G$ ,  $G_u = (V, E_u)$ , where  $V$  is the vertex set of the original TSP instance and  $E_u$  is the set of uncommon edges found in  $E$ . Typically,  $G_u$  is made up of multiple disconnected subgraphs. We use Breadth First Search on  $G_u$  to find each subgraphs (e.g., connected components) of  $G_u$ ; this has  $O(n)$  cost, because the degree of any vertex is at most 4, and each vertex is processed only once. Some additional bookkeeping is needed to track which partitions have cost 2 for graph  $G$ . When all the partitions of cost 2 are applied, the graph  $G$  is broken into  $k$  pieces which we will define to be *partition components*; not all connected components in  $G_u$  yield feasible partition components because they may not yield partitions of cost 2.

Figure 1(b) shows an example of the graph  $G_u$  created from the graph  $G$  shown in Figure 1(a). GPX creates children tours by using the common edges from  $G$  and taking either the dashed or solid edges from each of the partition components. The path followed by a tour within a partition component is independent of the path followed by a tour in any other partition component. Thus, within each component, a new tour could follow the path of the “dashed” parent or the “solid” parent.

**The GPX Theorem:**

Let graph  $G$  be constructed by unioning the vertices and edge found in two Hamiltonian Circuits for some instance of the TSP. If graph  $G$  can be separated into  $k$  partition components of cost 2, then there are  $2^k - 2$  distinct offspring; every recombination is both respectful and transmits alleles.

**Proof:**

By construction, all of the common edges connecting partition components are inherited. Within a partition component, a path followed by one of the parents is followed. This also means that all common edges within a partition component are inherited. Therefore the operator is respectful: all common edges are inherited. Generalized Partition Crossover “transmits alleles” because it only uses edges found in the graph  $G$ .

Let  $s$  be a string of  $k$  bits, one bit for each partition component. Let bit  $s_i = 0$  if a tour follows the path of the “dashed” parent in partition component  $i$ . Let bit  $s_i = 1$  if a tour follows the path of the “solid” parent in partition component  $i$ . Clearly, there are  $2^k$  possible tours (and bit strings) that can be constructed by GPX, but 2 of these represent the parents. Thus, if there are  $k$  partition components, there are  $2^k - 2$  possible offspring tours that are respectful and that transmit alleles.  $\square$

Since the objective function is linear, and all of the offspring are tours, if we make a greedy choice in each partition component by deciding whether the dashed-path or solid-path is shortest, we can also construct the shortest tour possible of the  $2^k - 2$  possible offspring by making  $k$  greedy choices within each partition component. This is also accomplished in  $O(n)$  time.

“Respect” means that children inherit the common edges; *thus, partitions are also always inherited*. This means we could recombine the parents using some partition A, then recombine the children again using some partition B to produce grandchildren. Thus, Generalized Partition Crossover is equivalent to iterative (i.e., multiple) applications of Partition Crossover.

## 2.1 GPX with Local Search

We run local search to generate an initial population and to further optimize the offspring. We tested different local search operators. In our previous study [2] looking at randomly chosen local optima produced by the application of 2-opt, we found that Partition Crossover was feasible more than 90 percent of the time. Since PX and GPX are closely related operators, when one is feasible the other is feasible. When randomly chosen local optima are generated by the application of 3-opt, we found that Generalized Partition Crossover is feasible in 100 percent of the cases tested across all of the TSP instances studied in this paper; in more than half of all cases, the offspring are still locally optimal under 3-opt.

To ascertain the number of partition components available to GPX, we recombined 50 random local optima generated using 2-opt [3], 3-opt and Lin-Kernighan search [4] (LK-search). The results are presented in Table 1. The instances rand500 and rand1500 are random Euclidean instances and att532,

| Instance  | rand500        | att532         | nrv1379        | rand1500       | u1817          |
|-----------|----------------|----------------|----------------|----------------|----------------|
| 2-opt     | $2.6 \pm 0.1$  | $3.3 \pm 0.2$  | $3.2 \pm 0.2$  | $3.7 \pm 0.3$  | $5.0 \pm 0.3$  |
| 3-opt     | $9.42 \pm 0.4$ | $10.5 \pm 0.5$ | $11.3 \pm 0.5$ | $24.9 \pm 0.2$ | $26.2 \pm 0.7$ |
| LK-search | $4.5 \pm 0.2$  | $5.3 \pm 0.2$  | $5.2 \pm 0.3$  | $10.6 \pm 0.3$ | $13.3 \pm 0.4$ |

**Table 1.** Average number of *partition components* used by GPX in 50 recombinations of random local optima found by 2-opt, 3-opt and LK-search.

nrv1379 and u1817 are from the TSPLIB. The number of cities in each instance is indicated by the numerical suffix.

Our data indicate 3-opt induces more partition components than 2-opt because it induces more subtours made up of common edges that can be used to partition the graph. The big valley hypothesis [5] supports a strong inverse correlation between the number of common edges shared with the global optimum and the evaluation of a tour. Tours produced by LK-search have even more *total* common edges than 3-opt, but some partitions are now absorbed into common subtours. Thus, there are a smaller number of common subtours (and fewer partitions) than 3-opt but the common subtours become longer for LK.

Sometimes there are more than 20 partition components under 3-opt. Using more than 20 partition components, one recombination is selecting the best of more than 1 million solutions, most of which are local optima. Nevertheless, working with LK-search gets us closer to the global optimum. In the remainder of this paper we will only employ LK-search.

### 3 The Algorithms: Descriptions and Comparisons

Our hybrid GA is described in Figure 2. The initial population is produced by randomly generating  $t$  tours and applying the same LK-search procedure used in step 5. The algorithm was run for a fixed number of generations.

The version of LK-search used is the implementation from Applegate et al. [6] with don't-look bits and uses the default neighborhood list size and search depth. By restricting moves (and clever programming), one iteration of LK-search is faster than a full exploration of the 2-opt neighborhood. The ordering and choice of neighbors is non-deterministic, meaning LK-search may improve upon a tour with subsequent calls until the potential improving moves under LK-search have been exhausted. Only one call to LK-search is done in Step 5.

One of the first questions we considered was whether to use truncation selection (always keep the  $t$  best solutions), or to try to preserve diversity by keeping offspring containing edges that are under-represented in the population. When GPX is applied in a greedy fashion, we generate two offspring. The first offspring is the greedy offspring: the shortest path in each partition component is selected. Usually, there are many small partition components, and one large partition component that is typically 20% of the tour. The second offspring also inherits the shortest path in all of the partition components, except for the largest partition

|  |
|--|
| Let $P1$ be a randomly generated population of size $t$ ;<br>Let $P2$ be a temporary child population of size $t$ ;<br>For each member of $P1$ : apply LK-search and evaluate;   |
| <ol style="list-style-type: none"> <li>1. Attempt to recombine the best tour of <math>P1</math> with the remaining <math>t - 1</math> tours using GPX; this generates a set of up to <math>2t</math> offspring.</li> <li>2. If recombine was not feasible between the best tour and tour <math>i</math>, mutate tour <math>i</math> using a double bridge move and place in population <math>P2</math>;</li> <li>3. Place the best solution found so far in population <math>P2</math>;</li> <li>4. From the set of offspring, select offspring to fill population <math>P2</math>;</li> <li>5. For each member of population <math>P2</math>: apply LK-search and evaluate;</li> <li>6. <math>P1 = P2</math>; If stopping condition not met, goto 1.</li> </ol> |

**Fig. 2.** Algorithm for the hybrid GA; the GA is generational, but elitist.

component; in this component the offspring inherits the path *not* used by the first greedy offspring. Thus,  $t$  recombinations produces  $2t$  offspring.

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. In empirical studies, a generational GA using diversity selection consistently produced much better results than keeping the  $t$  tours with lowest cost. Thus, we used “diversity selection” (step 4) in the Hybrid GA.

We retain the *best* tour found so far (step 3) in the population of offspring. If GPX fails to recombine two tours, we then 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).

### 3.1 Comparisons: The Hybrid GA with GPX versus Chained-LK

The hybrid GA was run using a population of 10 tours. The hybrid GA used LK-search as the local search method and Generalized Partition Crossover. We then compare the minimum tour found using the Hybrid Genetic Algorithm against the best tour found using Chained Lin-Kernighan. Both methods used exactly the same implementation of LK-search using identical parameters.

Chained Lin-Kernighan is one of the best performing local search heuristics for the TSP [6]. Chained LK applies LK-search to a single tour and then uses

| Generation → |            | 5           | 10           | 20           | 50           |
|--------------|------------|-------------|--------------|--------------|--------------|
| Instance     | Algorithm  | 60 LK calls | 110 LK calls | 210 LK calls | 510 LK calls |
| rand500      | GA w/ GPX  | 0.29 ± 0.01 | 0.17 ± 0     | 0.1 ± 0      | 0.05 ± 0     |
|              | Chained-LK | 0.30 ± 0.01 | 0.19 ± 0.01  | 0.13 ± 0     | 0.09 ± 0     |
| att532       | GA w/ GPX  | 0.29 ± 0    | 0.18 ± 0     | 0.12 ± 0     | 0.07 ± 0     |
|              | Chained-LK | 0.30 ± 0.01 | 0.21 ± 0.01  | 0.13 ± 0     | 0.08 ± 0     |
| nrw1379      | GA w/ GPX  | 0.63 ± 0    | 0.48 ± 0     | 0.34 ± 0     | 0.23 ± 0     |
|              | Chained-LK | 0.62 ± 0.01 | 0.46 ± 0.01  | 0.32 ± 0     | 0.19 ± 0     |
| rand1500     | GA w/ GPX  | 0.71 ± 0.01 | 0.52 ± 0.01  | 0.36 ± 0     | 0.22 ± 0     |
|              | Chained-LK | 0.73 ± 0.01 | 0.54 ± 0.01  | 0.39 ± 0.01  | 0.25 ± 0     |
| ul1817       | GA w/ GPX  | 1.61 ± 0.01 | 1.26 ± 0.01  | 0.95 ± 0.01  | 0.63 ± 0.01  |
|              | Chained-LK | 2.08 ± 0.02 | 1.61 ± 0.02  | 1.19 ± 0.01  | 0.83 ± 0.01  |

**Table 2.** Average percentage of the cost of the minimum tour found above the globally optimal cost averaged over 500 experiments using Chained LK and the Hybrid GA.

a double bridge move [7] to perturb the solution; Chained-LK then reapplies LK-search. Since the population size is 10, the hybrid GA uses 10 applications of LK-search each generation; therefore, Chained LK is allowed to do 10 double-bridge moves and apply the LK-search 10 times for every generation that the Hybrid GA is allowed to execute. This means that each algorithm is allowed to call LK-search exactly the same number of times. The Hybrid GA has the additional cost of recombination, but this cost is  $O(n)$  with a small constant (of 4 or 5) and the computation is very small compared to one iteration of LK-search. Furthermore, applying LK-search after a double bridge move is more expensive than applying LK-search after recombination: the solutions are made poorer by the double bridge move, and improved by recombination. Thus the run times are approximately the same. (Exact comparisons of run times are difficult because LK-search is integrated into the Chained-LK code, while the Hybrid GA uses a simple but unoptimized interface to call the LK-search.)

Table 2 lists the average percentage of the cost of the minimum tour found compared to the cost of the global optimum for each problem instance. The hybrid GA was allowed to run for 50 generations in these experiments.

After 510 calls each to LK-search, the Hybrid GA using GPX yields better results on all of the problems except nrw1379. This is remarkable because the Hybrid GA must optimize 10 solutions and the best solution must be optimized 10 times faster than Chained-LK to obtain a better result.

If each algorithm is run longer, the performance of the Hybrid Genetic Algorithm is increasingly better than Chained LK. Table 3 shows how many times (out of 50 attempts) that each method finds the global optimum after 1010 calls to LK-search (which is 100 generations for the Hybrid GA).

|            | rand500 | att532 | nrw1379 | rand1500 | u1817 |
|------------|---------|--------|---------|----------|-------|
| Hybrid GA  | 50/50   | 26/50  | 1/50    | 12/50    | 1/50  |
| Chained-LK | 38/50   | 16/50  | 1/50    | 2/50     | 0/50  |

**Table 3.** The number of times the global optimum is found by each algorithm after 1010 calls to LK-search over 50 experiments.

| Instance | Global Edges<br>in Population | Global Edges<br>in Minimum Tour | Unique Edges<br>in Population |
|----------|-------------------------------|---------------------------------|-------------------------------|
| rand500  | 500 $\pm$ 0                   | 449.68 $\pm$ 1.98               | 941.56 $\pm$ 1.56             |
| att532   | 532 $\pm$ 0                   | 464.1 $\pm$ 2.11                | 979.54 $\pm$ 1.47             |
| nrw1379  | 1378.9 $\pm$ 0.04             | 1162.3 $\pm$ 3.44               | 2709.34 $\pm$ 2.25            |
| rand1500 | 1500 $\pm$ 0                  | 1301.02 $\pm$ 4.15              | 2871.9 $\pm$ 3.14             |
| u1817    | 1815.12 $\pm$ 0.18            | 1562.44 $\pm$ 3.22              | 3616.92 $\pm$ 4.71            |

**Table 4.** Results obtained by running the hybrid GA for only 5 generations and *without* mutation (double-bridge moves). “Global Edges in population” is the number of edges found in the global optimum that are also present in the population. “Global Edges in Minimal Tour” is the number of edges shared in common between the best solution found and the global optimum. “Unique Edges in Population” is the total number of distinct edges found in the population at the end of generation 5. The number of edges in each instance is indicated by the numerical suffix of the instance name

## 4 The Power of a Population

While it is encouraging that the Hybrid GA is able to yield performance that exceeds that of Chained-LK, is this really the best way to exploit the population of solutions that is being generated by the Hybrid GA? To explore this question, we ran the Hybrid GA again. However, we turned off the mutation operator in step 2. This means that step 4 now selects  $t - 1$  offspring to place in population  $P_2$ . During the first few generations, recombination is almost always feasible.

We ran 50 trials of the Hybrid GA for only 5 generations. At generation 5 we record the minimum tour found, the number of unique edges in the population and the number of edges in the population that also appear in the global optimum. When there is no mutation (i.e., double bridge moves) the Hybrid GA converges very fast, but it also loses diversity and gets “stuck” after about 5 generations.

Nevertheless, the Hybrid GA is already finding very good solutions after only 5 generations: for rand500 and att532, it found the global optimum in 2 out of 50 trials, using only 50 recombinations and only 50 calls to LK-search. The convergence to the global optimum is extremely fast in these exceptional cases.

As the data shows in table 4, the edges found in the global optimum are all present in the population in the majority of the runs. On instances rand500, att532 and rand1500 all of the edges found in the global optimum were also in

the population on every single run. The population therefore contains all of the edges needed to construct the globally optimal solution after only 5 generations.

Furthermore, the results for all of the TSP instances show that the total number of unique edges in the population was always less than  $2n$  after 5 generations. Assume that we merge all 10 members of the population after 5 generations into a single graph. We can now search this reduced graph for a minimal Hamiltonian circuit. The search space is dramatically smaller than that of the original TSP instances. This means that the optimization problem has been reduced to finding the minimal Hamiltonian Circuit of length  $n$  in a graph with only  $2n$  edges.

#### 4.1 Where are the Global Edges?

We next look at those edges that appear in the global optimum, but which do not appear in the best tour in the population. We already know that typically all of the edges found in the global optimum are present in the population after 5 generations. Since we recombine the best tour with all members of the population, those edges that are shared with the global but not found in the minimal tour must be classified as uncommon by GPX and will appear in the graph  $G_u$  during at least one recombination.

Because of the way GPX performs crossover, edges that appear in the same partition component cannot be chosen on an individual level. Either all the edges from one parent are chosen from that component or all the edges from the other parent are chosen.

We want to determine if the uncommon globally optimal edges are spread out among different partition components in  $G_u$  or if they appear in the same component. If a majority of the uncommon global edges appear in the same component, then this means it will be impossible for GPX (working without any form of mutation) to reassemble these edges and reach the global optimum.

We looked at the trials from the previous experiments and found in the majority of recombinations the uncommon globally optimal edges fell into a single partition component which was larger than the rest. In table 5 we report the number of uncommon globally optimal edges which fell into this large partition component, the size of this component, and the number of uncommon global edges which fell into other components; we also report the number of shared common global edges observed during recombination. Results for two instances, att532 and u1817, are shown in table 5.

As can be seen from Table 5, the majority of edges that are not found in the best solution but that are found in the global optimal solution appear as uncommon edges that are largely contained in the largest component during the recombination process. Nevertheless, most of the edges that are found in the globally optimum actually appear as *common* edges (the first row in Table 5) during recombination, meaning these edges will be passed onto the children.

|  | att532 | u1817   |
|--|--------|---------|
| Common Edges also found in the global optimum                    | 406.75 | 1407.90 |
| Uncommon edges in largest component also found in global optimum | 87.39  | 268.62  |
| Uncommon in all other components also found in global optimum    | 0.04   | 0.96    |
| Total edges in the largest component                             | 102.11 | 289.12  |

**Table 5.** Percentages were averaged over 50 trials. These results were captured during recombination during the 5th generation. Common Edges can appear inside of components, or between components of Graph  $G_u$ . Uncommon edges appear only inside of components of  $G_u$ .

## 5 Conclusions and Future Work

A new recombination operation has been developed for the TSP.GPX is a generalization of the previously described PX operator. Both operators are respectful and transmits alleles. GPX in a hybrid GA with LK-search is capable of finding better tours than Chained LK using double bridge moves. Additionally, we find that the Hybrid Genetic Algorithm using GPX and LK-search is capable of finding globally optimal solutions in a relatively small number of generations.

Additional analysis shows that all the edges found in the globally optimal solution are present in a population after only a few generations in almost every case. Furthermore, the number of unique edges in the population is also less than twice the problem size.

When critical edges are concentrated in a single partition component, GPX is not able to “re-assort” these edges. However, this represents a challenge as well as an opportunity. Instead of needing to optimize over the entire search space, effort can be focused on optimizing a small subregion of the search space. Future research will examine how best to exploit this knowledge.

## References

1. Radcliffe, N., Surry, P.: Fitness variance of formae and performance predictions. In Whitley, D., Vose, M., eds.: FOGA - 3, Morgan Kaufmann (1995) 51–72
2. Whitley, D., Hains, D., Howe, A.: Tunneling between optima: partition crossover for the traveling salesman problem. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM (2009) 915–922
3. Croes, G.: A method for solving traveling-salesman problems. *Operations Research* (1958) 791–812
4. Lin, S., Kernighan, B.: An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* (1973) 498–516
5. Boese, K.D., Kahng, A.B., Muddu, S.: A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* **16** (1994) 101–113
6. Applegate, D., Cook, W., Rohe, A.: Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing* **15**(1) (2003) 82–92
7. Johnson, D.S., McGeoch, L.A.: The traveling salesman problem: A case study in local optimization. In Aarts, E.H.L., Lenstra, J., eds.: *Local Search in Combinatorial Optimization*. John Wiley and Sons Ltd (1997) 215–310