

# Improving Edge Recombination through Alternate Inheritance and Greedy Manner

Chuan-Kang Ting

International Graduate School of Dynamic Intelligent Systems  
University Paderborn, 33098 Paderborn, Germany  
ckting@upb.de

**Abstract.** Genetic Algorithms (GAs) are well-known heuristic algorithms and have been widely applied to solve combinatorial problems. Edge recombination is one of the famous crossovers designed for GAs to solve combinatorial problems. The essence of edge recombination is to achieve maximal inheritance from parental edges. This paper presents two strategies to improve edge recombination. First, we encourage alternation of parents in edge inheritance. Second, a greedy method is used to handle the failures occurred in edge recombination. A modified edge recombination, called edge recombination with tabu (Edge-T), is proposed according to these two strategies. The traveling salesman problem is used as a benchmark to demonstrate the effectiveness of the proposed method. Experimental results indicate that Edge-T can achieve better performance than the conventional edge recombination Edge-3 in terms of both solution quality and convergence speed.

## 1 Introduction

Genetic algorithms (GAs) are well-known heuristic algorithms based on the imitation of natural evolution [Hol75]. Their effectiveness in search and optimization problems has been extensively validated. Crossover is one of the most salient features of GAs. It simulates the creation of offspring in the natural world by exchanging and recombining genetic information from two selected parents. This operator is believed to be capable of exploring the problem space effectively. However, in order to enhance GAs' performance, several crossovers have been proposed to deal with specific problems. For example, in terms of combinatorial problems, there exist partially mapped crossover (PMX) [GL85], order crossover (OX) [Dav85], and cycle crossover (CX) [OSH87].

The traveling salesman problem (TSP) is a classic combinatorial problem. Given  $n$  cities, the objective of TSP is to find the shortest tour that visits all cities exactly once. This problem has been proven NP-complete; in other words, under the general assumption  $P \neq NP$ , there is no exact algorithm which can solve the problem in polynomial time. Heuristic algorithms, therefore, are commonly used to deal with this problem.

To solve the TSP with a GA, the crossovers used for order based representations, such as PMX, OX, and CX, are traditionally adopted. The character of these cross

overs is to permute and recombine the order information of parental chromosomes. The adjacency information, however, is crucial to TSP but is not utilized in these operators. Whitley et al. [WSF89] proposed a series of crossovers, called *edge recombination*, to overcome this drawback. Edge recombination utilizes an edge table to record the adjacency of parental cities, and then builds the filial tour with reference to this edge table and specific heuristics. The method (Edge-1) and its modifications (Edge-2,3,4,5,6) have shown their superiority over PMX, OX, and CX in TSP [Sta91, NYY00].

Generally, edge recombination attempts to inherit as many parental edges as possible. Its ability to preserve parental edges has been validated through its higher correlation coefficient between the fitness of parents and offspring [MdWS91]. This ability is believed useful to exploit the fitness landscape and is capable of better performance. However, another adjacency-based crossover, edge assembly crossover (EAX) [NK97, NK99] uses different strategies from such a “maximal inheritance” tactic. First, an *AB-cycle* is defined as a cycle formed by alternating parental sequences. Second, a so-called *E-set* collects a subset of AB-cycles by two methods: random selection and heuristic selection. The intermediate sub-tours are then established on the basis of the E-set. Finally, EAX uses a greedy manner to merge these sub-tours. This procedure attempts to connect two sub-tours with the edges which can achieve the minimal total distance of these two sub-tours. Experimental results show that EAX achieves a breakthrough in the performance of crossover in TSP [Wat98]

Considering EAX’s success we propose the following two strategies to enhance edge recombination.

1. Inheritance of edges from alternate parents.
2. Greedy manner in connecting sub-tours.

Instead of maximal inheritance, these two strategies result in more diverse inheritance. The alternation of parental inheritance can not only preserve parental information but also utilize this information in a more explorative way. The original edge recombination attempts to generate highly correlated offspring with their parents. This highly-correlated inheritance, nevertheless, poses a risk of the biased inheritance, which inherits edges merely from one parent and consequently renders the population similar. An alternation of parents can prevent this drawback and maintain population diversity. In addition, the concept of maximal inheritance also affects the failure treatment in edge recombination. The original edge recombination and its modifications all try to recover and extend the tour-building process when failure occurs. However, these deadlock tour segments can be viewed as sub-tours. Our second strategy, therefore, does not prevent failures from introducing foreign edges as the original edge recombination do. Instead, our proposed greedy manner attempts to introduce an *advantageous* foreign edge to connect these sub-tours.

In this paper, a modified edge recombination, called *edge recombination with tabu* (Edge-T), is proposed to validate the effectiveness of these two strategies in edge recombination. First, a tabu list [GK97] is applied to Edge-T in order to restrain inheriting the successive edge from the same parent. Such a restriction is expected to increase the alternate level of parents. Second, Edge-T adopts a greedy method rather

than a recovery manner to handle the failure condition. Several TSP instances, ranging from 51 to 442 cities, are used as benchmarks to evaluate the performance of Edge-T.

The rest of this paper is organized as follows. Section 2 gives a brief description of edge recombination. Next, Section 3 describes the algorithm Edge-T and its strategies in a more detail. Section 4 presents experimental results including the influence of Edge-T's components and performance comparisons. Finally, conclusions are drawn in Section 5.

## 2 Edge Recombination

Edge recombination is a crossover which focuses on the adjacency relation. As illustrated in Figure 1, edge recombination uses an edge table to record parental edges, and then limits the generation of offspring to the edges contained in this table. In other words, the candidates of offspring edges come from parental edges principally.

With reference to the edge table, edge recombination builds a tour according to specific heuristics. In the original edge recombination (Edge-1) [WSF89], the building process intends to select the city with the fewest links, namely, the most *isolating* city. Edge-1 initially generates the edge table by scanning both parents. Afterwards, Edge-1 begins the process of building the filial tour. (1) Select one of the first parental cities as the starting city. (2) Select its next city from the remaining adjacent cities (links) of the current city. We call these links *candidates* for the next city. According to the heuristic's "priority of isolating cites", the candidates with the smallest number of links is chosen. (3) Repeat (2) until the complete tour is built.

City	Links	#Links	City	Links	#Links
A	B, J, H	3	F	E, G, C	3
B	A, C, I, H	4	G	F, H, C, I	4
C	B, D, F, G	4	H	G, I, B, A	4
D	C, E, J	3	I	H, J, G, B	4
E	D, F	2	J	I, A, D	3

**Fig. 1.** Edge table of edge recombination

Several modifications based on different heuristics have been proposed to improve the original edge recombination. Edge-2 [Sta91], for instance, pays more attention to common edges of both parents than the isolating cites. It marks the common link with a negative (-) sign to indicate its priority over other odd links. Experimental results demonstrate this heuristic can improve the original edge recombination. In addition to the process of building a tour, some modifications focus on handling the failure condition. In edge recombination a failure occurs when the crossover can not find any possible candidate for the next city. The original edge recombination solves this deadlock by randomly choosing the next city from the unselected cities. In this case, the crossover is forced to introduce a *foreign* edge, which means an edge that does not exist in parental edges. Such a foreign edge is believed to be harmful to the perform-

ance of crossover. In order to reduce the number of failures, Edge-3 [MW92] reverses the built tour segment when a failure occurs, and then continues the building process from the *live* end instead of the *dead* end. If another failure occurs thereafter, the algorithm will randomly choose the next city as Edge-1 does because there is no *live* end for this built tour segment. This randomly chosen city subsequently acts as a beginning of a new tour segment and is viewed as another *live* end. Edge-4 [DW94], nonetheless, only reverses a partial segment rather than the whole built segment. Furthermore, Edge-5 [NYY00] employs another heuristic of choosing the next city to limit the number of failures. Edge-5 intends to select the edges from the same parent during building tour. The results show this approach can significantly reduce the failure rate at the mature stage. Edge-6 [NYY00] further deals with the failures in Edge-5. When a failure occurs, Edge-6 examines if there is an edge connecting two ends of the built tour segment. If it holds, Edge-6 will seek a city along the built tour to link an unselected city. Then Edge-6 breaks the connection of the sought city and its successive city, and re-connects it with the unselected city. Experimental results indicate Edge-5 and Edge-6 can decrease the failure rate and obtain better performance.

### 3 Edge Recombination with Tabu (Edge-T)

The *edge recombination with tabu* (Edge-T) is proposed to validate our two strategies in enhancing edge recombination. First, we argue that the alternation of parents in inheritance is capable of better performance. To meet this purpose, Edge-T incorporates a tabu list to prevent inheriting the successive edge from the same parent. Second, we consider a greedy manner in handling failures can improve the performance. Here a built tour segment that encounters a failure is viewed as a sub-tour. Edge-T attempts to introduce the shortest foreign edge to connect these sub-tours.

To incorporate the first strategies with edge recombination, Edge-T uses a priority function to sum up the weights of common edges and isolating level. Furthermore, the factor of tabu restriction is added to this priority function as a penalty. The function used to evaluate the *priority* of an edge from the current city  $i$  to its link  $j$  is thus defined:

$$P_{ij} = 2m_{ij} + (4 - l_j) - T_{ij} \quad (1)$$

where  $m_{ij}$  denotes the number of common edges,  $l_j$  denotes the number of remaining links for city  $j$ , and  $T_{ij}$  is the number of tabu occurrences. Edge-T always chooses the link with the greatest priority value. The first term of the priority function indicates the portion of common edges. In the priority function only the value  $m_{ij}$  greater than 1 will be taken into calculation; otherwise this value is set to 0. The second term expresses the portion of isolating level. The number of remaining links  $l_j$  ranges from 0 to 4 and is inversely proportional to the isolating level. To meet the *maximization* of the priority function, we subtract the value  $l_j$  from its maximal value 4. The last term,  $T_{ij}$ , accounts for the penalty factor. A *tabu list* [GK97] is applied to record the prohibitive parents for inheritance. In addition, a corresponding *ancestry list* is necessarily added to each link to trace the parents where the edge inherits from.

As Figure 2 shows, for example, in terms of city A, the link to city B comes from parent 1; the link to city J is a common edge of parent 1 and parent 2. When an edge is selected from the edge table, the ancestry list of this edge turns into the tabu list, which inhibits the building process to choose the next edge from the same parent.

Edge Table:

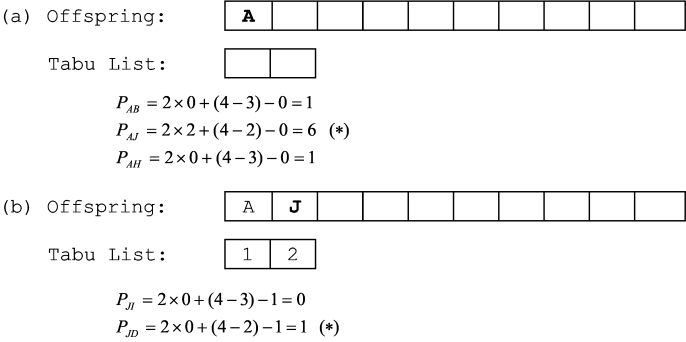
City	Links	#Links	City	Links	#Links
A	B (1), J (1, 2), H (2)	3	F	E (1, 2), G (1), C (2)	3
B	A (1), C (1), I (2), H (2)	4	G	F (1), H (1), C (2), I (2)	4
C	B (1), D (1), F (2), G (2)	4	H	G (1), I (1), B (2), A (2)	4
D	C (1), E (1, 2), J (2)	3	I	H (1), J (1), G (2), B (2)	4
E	D (1, 2), F (1, 2)	2	J	I (1), A (1, 2), D (2)	3

Tabu List:

1	
---	--

**Fig. 2.** The extended edge table and tabu list of Edge-T

Totally speaking, these three portions – common edges, isolating level, and tabu restriction – constitute the decision criteria for building tours. Following the previous example, if the offspring tour starts with city A, it will have three available links to consider. The tabu list is empty at this time because there is no previous edge. Figure 3(a) shows the priority of its candidates B, J, H is 1, 6, and 1 respectively. Due to the greatest priority value, city J is picked and then its ancestry list turns into the tabu list as shown in Figure 3(b). This “evaluate-and-choose” process will continue until the offspring tour is finished or a failure occurs.



**Fig. 3.** Example of 10-city tour building in Edge-T

Failures, nevertheless, occur probably in Edge-T as well as in edge recombination. Unlike edge recombination’s preventive way, Edge-T adopt a greedy manner to introduce foreign but advantageous edges instead. When a failure occurs, namely no available edge to choose, the built tour segment can be viewed as a *sub-tour*. During the building process, there may be some of such sub-tours. In this case, Edge-T will seek the *shortest* link between two sub-tours. Specifically, Edge-T will search the unse-

lected cities for the shortest link from the end of currently built segment. The complexity of this search process is  $O(n)$ , where  $n$  is the number of cities in TSP. This computation cost is not so expensive but will bring a considerable advantage to the performance of crossover.

## 4 Experiments

In this paper, a series of experiments are conducted to evaluate the performance of Edge-T. These experiments adopt eight TSP instances [TSPLIB] which are widely used as benchmarks. Table 1 lists these experimental instances and their respective optimal tour length.

The schemes of GA employed in our experiments are steady-state GA, path representation, linear-ranking selection with bias 1.25. Here we set complete crossover ( $p_c = 1.0$ ) and no mutation ( $p_m = 0$ ) to examine the absolute influence of crossover on performance, although the utilization of mutation may result in better solutions. The survival strategy is to delete the worst member of a family, which consists of two parents and their children. A population size of 500, 1000, and 2000 chromosomes is empirically set for problems fewer than 300 cities, lin318, and pcb442 respectively. Each experimental setting includes 30 independent runs where population is initialized randomly.

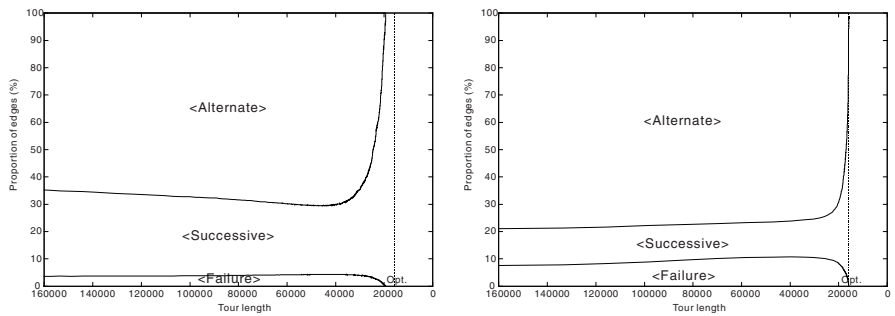
The experiments first examine the impact of tabu list upon the alternation of parental edges. Second, convergence comparisons are presented to verify the performance of Edge-T in subsection 4.2.

### 4.1 Influence of Tabu List

Edge-T applies a tabu list to restrain the edge inheritance from the same parent. We examine the effectiveness of this tabu restriction with an inspection upon the composition of a tour. Here only the results on the TSP instance d198 (198 cities) is presented because of the similar results as other instances. Figure 4 shows the proportions of edges from alternate parents, successive parents, and failure treatment. The result indicates that Edge-T has a steadily higher proportion of alternate parents than Edge-3. This outcome validates the effectiveness of tabu restriction in alternating parents. Its corresponding influence on convergent performance will be further checked in the next subsection. In addition, Edge-T has a higher failure proportion than that of Edge-3. This higher failure rate of Edge-T reveals that the failure treatment plays an important role in the performance.

### 4.2 Performance Comparison

The effectiveness of Edge-T on solution quality and convergence speed is further examined by comparison with Edge-3, Edge-Tx (Edge-T without tabu restriction), and Edge-Ty (Edge-T without tabu restriction and using the failure treatment of Edge-3). Table 1 and Table 3 summarize the experimental results on TSP instances ranging from 51 to 442 cities.



**Fig. 4.** Proportion of edges from alternate parents, successive parents, and failure treatment of Edge-3 (left) and Edge-T (right).

Table 1 compares the solution quality. Edge-T achieves the best solution quality on average and the smallest standard deviation on all experiments. Edge-Tx, meanwhile, obtains better solution and smaller standard deviation than Edge-3 and Edge-Ty does. However, there is no clear difference between Edge-3 and Edge-Ty. A series of one-tailed t-tests is further performed to validate the significance of superiority in terms of solution quality. Table 2 presents the statistical results of t-test on the experimental data of Table 1. With a confidence level  $\alpha = 0.01$ , the optimal solutions obtained from Edge-3 and Edge-Ty do not differ significantly on most problems except lin318 and pcb442. It indicates the priority function of Edge-T does not result in significant difference in performance from Edge-3. In addition, the statistical results  $p < 0.01$  in the (EdgeTy – EdgeTx) and (Edge3 – EdgeTx) column validate that Edge-Tx can achieve better solution than Edge-Ty and Edge-3. These two outcomes support that the greedy manner can improve the failure treatment of Edge-3. Furthermore, the proposed algorithm Edge-T, achieves even better solution quality than Edge-Tx according to the t-test results in the (EdgeTx – EdgeT) column of Table 2. This preferable outcome confirms the effectiveness of tabu restriction on the advanced improvement of Edge-Tx. As shown in Figure 4, the tabu restriction results in not only a larger-scale parental alternation but also a higher failure rate. The superior performance of Edge-T, therefore, is attributed to a joint effort of these two effects.

**Table 1.** Comparative tour length of Edge-3 and Edge-T series

TSP instance	Opt	Edge-3		Edge-Ty		Edge-Tx		Edge-T	
		Ave	Std	Ave	Std	Ave	Std	Ave	Std
eil51	426	430.0	3.2	430.6	3.9	427.8	2.2	427.0	0.6
gr96	51229	54456	1190	54448	1019	51915	445	51288	121
eil101	629	673.8	10.8	676.4	10.0	641.0	5.0	630.3	2.0
lin105	14379	15643	478	15795	514	14717	236	14477	53
d198	15780	19397	506	19366	403	16300	228	15879	38
kroA200	29368	39327	1114	40006	1216	30457	313	29657	167
lin318	42029	93830	12548	80808	13552	44156	2470	43371	873
pcb442	50778	171426	11345	155100	9276	62914	2964	59071	2471

**Table 2.** t-test of solution quality in term of tour length

TSP in- stance	Edge3 – EdgeTy		Edge3 – EdgeTx		EdgeTy – EdgeTx		EdgeTx – EdgeT	
	<i>t(df)</i>	<i>p</i>	<i>t(df)</i>	<i>p</i>	<i>t(df)</i>	<i>p</i>	<i>t(df)</i>	<i>p</i>
eil51	-0.65 (55)	0.259	3.10 (51)	1.56E-03	3.43 (45)	6.61E-04	1.92 (33)	3.17E-02
gr96	0.03 (56)	0.489	10.96 (36)	2.57E-13	12.48 (39)	1.71E-15	7.45 (33)	7.36E-09
eil101	-0.97 (57)	0.169	15.09 (40)	2.06E-18	17.34 (42)	4.65E-21	10.88 (38)	1.54E-13
lin105	-1.19 (57)	0.120	9.52 (42)	2.43E-12	10.44 (40)	2.75E-13	5.43 (31)	3.09E-06
d198	0.26 (55)	0.397	30.59 (40)	1.19E-29	36.27 (45)	2.94E-35	9.98 (30)	2.42E-11
kroA200	-2.26 (57)	0.014	42.00 (33)	1.59E-30	41.65 (32)	9.44E-30	12.35 (44)	3.38E-16
lin318	3.86 (57)	1.45E-04	21.27 (31)	2.35E-20	14.57 (30)	1.88E-15	1.64 (36)	5.47E-02
pcb442	6.10 (55)	5.50E-08	50.69 (32)	1.94E-32	51.85 (34)	2.99E-34	5.45 (56)	5.75E-07

*df* : degree of freedom

Next, Table 3 compares the convergence speed in terms of the number of evaluations. Here the number of evaluations is defined as the number of reproduction, which is also equal to the number of crossovers. From Table 3, we can find Edge-T and Edge-Tx converges with fewer evaluations than Edge-3 and Edge-Ty; furthermore, Edge-T uses much fewer evaluations than Edge-Tx does. In most cases, Edge-T uses less than half number of evaluations that Edge-3 and Edge-Ty needs. Additionally, Edge-T gives a smaller standard deviation in the number of evaluations than other approaches. The one-tailed t-test is also given in Table 4 to test the significance of improvement in convergence speed. Here we still follow the confidence level  $\alpha = 0.01$ . The results of the test (Edge3 – EdgeTy) in Table 4 indicate Edge-Ty can converge faster than Edge-3 does in problem gr96, lin105, and kroA200, but this superiority does not exist in other test instances. In addition, the test (Edge3 – EdgeTx) and (EdgeTy – EdgeTx) shows that Edge-Tx is capable of a faster convergence than Edge-3 and Edge-Ty, which demonstrate the benefit of greedy manner. Finally, the test (EdgeT – EdgeTx) confirms that Edge-T can converge further faster than EdgeTx. This favorable outcome proves the utility of tabu restriction in accelerating convergence speed.

**Table 3.** Comparative number of evaluations

TSP in- stance	Edge-3		Edge-Ty		Edge-Tx		Edge-T	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
eil51	39599	3768	39730	4282	31788	4287	17766	2306
gr96	103932	8967	98328	7247	61236	5882	33811	4607
eil101	119691	18883	112961	15579	76612	10092	51660	7498
lin105	147018	13249	137509	16380	83175	11424	42045	3985
d198	653707	95779	608002	83460	284786	38178	171094	27040
kroA200	474843	57895	431967	59104	196572	19833	123077	12613
lin318	1666891	379840	1707216	411641	1473512	239780	784920	140114
pcb442	2910549	952821	3110969	979338	2423476	738469	1534216	851173



**Table 4.** t-test of convergence speed in terms of number of evaluations

TSP in- stance	Edge3 – EdgeTy		Edge3 – EdgeTx		EdgeTy – EdgeTx		EdgeTx – EdgeT	
	$t(df)$	$p$	$t(df)$	$p$	$t(df)$	$p$	$t(df)$	$p$
eil51	-0.13 (57)	0.450	7.50 (57)	2.37E-10	7.18 (57)	8.00E-10	15.78 (44)	4.99E-20
gr96	2.66 (55)	0.005	21.81 (50)	1.70E-27	21.77 (55)	5.27E-29	20.10 (54)	4.84E-27
eil101	1.51 (55)	0.069	11.02 (44)	1.53E-14	10.73 (49)	9.35E-15	10.87 (53)	2.10E-15
lin105	2.47 (55)	0.008	19.99 (56)	1.83E-27	14.90 (51)	1.61E-20	18.62 (35)	4.86E-20
d198	1.97 (56)	0.027	19.60 (37)	1.97E-21	19.29 (40)	3.63E-22	13.31 (52)	1.11E-18
kroA200	2.84 (57)	0.003	24.91 (35)	3.72E-24	20.68 (35)	1.66E-21	17.13 (49)	1.26E-22
lin318	-0.39 (57)	0.347	2.36 (48)	1.12E-02	2.69 (46)	5.00E-03	13.58 (46)	5.23E-18
pcb442	-0.80 (57)	0.213	2.21 (54)	1.56E-02	3.07 (53)	1.69E-03	4.32 (56)	3.20E-05

## 5 Conclusions and Future Work

In this work, we propose two strategies to improve the performance of edge recombination. First, inheritance of edges from alternate parents can enhance the performance of crossover. Second, a greedy manner in connecting sub-tours contributes to better results. The proposed algorithm Edge-T, based on edge recombination, is designed to verify these two strategies. The first strategy is accomplished through a penal way. Edge-T uses a tabu list to record the preceding parent and prohibit successive edge from inheriting the same parent. In addition, a greedy manner based on the second strategy is used to handle failures in Edge-T. The failure in edge recombination occurs when there is no city available to choose. This deadlock tour segment, however, can be viewed as a sub-tour. Accordingly Edge-T attempts to introduce the shortest edge to connect these sub-tours.

Several widely-used TSP instances sized from 51 cities to 442 cities are implemented to examine the performance of Edge-T. Experimental results demonstrate the capability that tabu restriction can increase the alternation of parents in inheritance. Furthermore, the performance comparisons confirm the two proposed strategies can significantly enhance Edge-T and result in better performance than Edge-3 in terms of both solution quality and convergence speed.

Much more work is needed to validate the effectiveness of the proposed strategies. First, the scale of TSP instances used in our experiments is relatively small. In addition, even though Edge-T can obtain better results than Edge-3 in these experiments, it does not achieve the optimal solution. We reason this defect due to the lack of local search. Genetic algorithms can efficiently and effectively lead the search to the promising region; however, without local search it becomes more than difficult to probe into the optimum, especially when the length of chromosome gets longer. Therefore, local search, such as 2-opt [Cro58] and Lin-Kernighan [LK73], is probably used to reinforce the search ability. Moreover, larger-scale TSP instances should be included in the benchmarks. Second, we disabled the mutation operator in our experiments for observing the behavior of crossover. This setting will cause the search to focus on exploitation, but on the other hand, exploration also needs to be considered. Hence, the operator which can enhance exploration, e.g. mutation or 2-opt, should be performed with crossover. Currently we are studying and conducting ex-

periments on these issues. More specific analysis on the impact of the two proposed strategies upon Edge-T and applications of these two strategies to other crossovers are also underway.

## References

- [Cro58] G.A. Croes: A Method for Solving Traveling Salesman Problems. *Operations Res.* 6 (1958) 791-812
- [Dav85] L. Davis: Applying Adaptive Algorithms to Domains. *Proc. Int. Joint Conf. on Artificial Intelligence* (1985) 162-164
- [DW94] J. Dzuberka and D. Whitley: Advanced Correlation Analysis of Operators for the Traveling Salesman Problems. *Parallel Problem Solving from Nature-PPSN III* (1994) 68-77
- [GL85] D. Goldberg and R. Lingle: Alleles, Loci and the Traveling Salesman Problem. *Proc. First Int. Conf. on Genetic Algorithms* (1985) 154-159
- [GK97] F. Glover, J.P. Kelly: *Tabu Search*. Kluwer Academic Publishers (1997)
- [Hol75] J.H. Holland: *Adaptation in Natural and Artificial Systems*. University of Michigan Press (1975)
- [LK73] S. Lin and B.W. Kernighan: An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Res.* 21 (1973), 498-516
- [MdWS91] B. Manderick, M. de Weger, P. Spiessens: The Genetic Algorithm and the Structure of the Fitness Landscape. *Proc. Fourth Int. Conf. on Genetic Algorithms* (1991) 143-150
- [MW92] K. Mathias and D. Whitley: Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem. *Parallel Problem Solving from Nature-PPSN 2* (1992) 219-228
- [NK97] Y. Nagata and S. Kobayashi: Edge Assembly Crossover: A High-Power Genetic Algorithm for the Traveling Salesman Problem. *Proc. Seventh Int. Conf. on Genetic Algorithms* (1997) 450-457
- [NK99] Y. Nagata and S. Kobayashi: An Analysis of Edge Assembly Crossover for the Traveling Salesman Problem. *Proc. IEEE Int. Conf. on Systems Man and Cybernetics* (1999) 628-633
- [NYY00] H.D. Nguyen, I. Yoshihara, and M. Yasunaga: Modified Edge Recombination Operators of Genetic Algorithms for the Traveling Salesman Problem. *Proc. IEEE Int. Conf. on Industrial Electronics, Control, and Instrumentation* (2000) 2815-2820
- [OSH87] I. Oliver, D. Smith, and J. Holland. A Study of Permutation Crossover Operators on the Traveling Salesman Problem. *Proc. Second Int. Conf. on Genetic Algorithms* (1987) 224-230
- [Sta91] T. Starkweather, S. McDaniel, K. Mathias, C. Whitley, and D. Whitley: A Comparative Study of Genetic Sequencing Operators. *Proc. Fourth Int. Conf. on Genetic Algorithms* (1991) 69-76
- [TSPLIB] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- [Wat98] J. P. Watson, C. Ross, V. Eisele, J. Denton, J. Bins, C. Guerra, D. Whitley, and A. Howe: The Traveling Salesrep Problem, Edge Assembly Crossover, and 2-opt. *Parallel Problem Solving from Nature-PPSN V* (1998)
- [WSF89] D. Whitley, T. Starkweather, and D. Fuquay. Scheduling Problems and the Traveling Salesman: the Genetic Edge Recombination Operator. *Proc. Third Int. Conf. on Genetic Algorithms and Their Applications* (1989)