Development of A Novel Crossover of Hybrid Genetic Algorithms for Large-scale Traveling Salesman Problems

Masafumi KURODA[1)], Kunihito YAMAMORI[1)], Masaharu MUNETOMO[2)], Moritoshi YASUNAGA[3)], Ikuo YOSHIHARA[1)]

[1)] *1-1, Gakuen Kibanadai-nishi, Miyazaki, 889-2192, Japan, University of Miyazaki*
(Tel : +81-985-58-7384; Fax : +81-985-58-7384)
(Email address : {kuroda, yamamori, yoshiha}@taurus.cs.miyazaki-u.ac.jp)

[2)] *Information Initiative Center, Hokkaido University*
[3)] *Graduate School of Systems and Information Engineering, University of Tsukuba*

Abstract: In this paper, we propose a novel crossover operator for solving the traveling salesman problem (TSP) with a Hybrid Genetic Algorithm (HGA) involving Lin-Kernighan (LK) heuristic for local search. We call the crossover operator Sub-tour Recombination Crossover (SRX) which divides each tour of the parents into many sub-tours under some rules and reconnects sub-tours from both the parents so as to construct a new tour of TSP. The method is evaluated from the viewpoint of tour quality and CPU time for ten well-known benchmarks e.g. dj38, qa194 … ch71009.tsp in the TSP website of Georgia Institute of Technology. We compare SRX with the conventional crossover operators; variant of the Maximal Preservative Crossover operator (MPX3), variant of the Greedy Sub-tour Crossover operator (GSX2) and variant of the Edge Recombination Crossover operator (ERX6), and show that the SRX succeeded in finding better solution and running faster than the conventional methods.

Keywords: Traveling Salesman Problem, Hybrid Genetic Algorithm, Lin-Kernighan, Crossover

I. INTRODUCTION

The traveling salesman problem (TSP) is one of the most widely studied problems in combinatorial optimization. Given a collection of cities and the cost of travel between each pair of cities, the TSP is to find the cheapest way of visiting all the cities only once and returning to the starting city. The TSP is NP-hard and very difficult to find true optimal solutions of large-scale instances. We will target instances of over 10^6 cities. We employ technique of a hybrid genetic algorithms (HGAs) combining GAs with Lin-Kernighan (LK) [1] heuristic for local search. The key of the method is a new crossover operator fitting to LK.

It is believed that to inherit long sub-tours from parents to offspring is important in order to avoid breeding a bad offspring. For example, in conventional crossovers, the Maximal Preservative Crossover (MPX) by Mühlenbein et al. and the Greedy Sub-tour Crossover (GSX) by Sengoku and Yoshihara inherit long sub-tours from both the parents to offspring [2][3]. The Edge Recombination Crossover (ERX) by Whitley et al. inherits many edges from both the parents to offspring [4]. Therefore the ERX inherits long sub-tours from them to offspring. But we believe that it is not always advantageous to inherit long sub-tours from both the parents to offspring when using heuristic. Because long sub-tours inherited from parents to offspring have already been improved by heuristic, there is almost no room to be improved by heuristic. However, if sub-tours are too short, they do not inherit parents' character well. Therefore, there must be an appropriate length of sub-tours cut out from the parents. We assume that length of sub-tours should be changed according to the problem size of the TSP. The preliminary experiments show that the best length of sub-tours is around $\sqrt{N}$ (N; number of cities).

We develop a novel crossover operator of HGA employing LK heuristic for local search. We call it Sub-tour Recombination Crossover (SRX) which divides each tour of the parents into many sub-tours that are limited by maximum tour length K, and reconnects sub-tours from both the parents so as to construct a new tour of TSP. We compare SRX with the conventional crossover operators; variant of the MPX operator (MPX3 [6]), variant of the GSX operator (GSX2 [7]) and variant of the ERX operator (ERX6 [8]). We used well-known 10 benchmarks with sizes from 38 to 71,009 cities in the TSP website of Georgia Institute of Technology. We evaluate the SRX from the viewpoint of tour quality and CPU time.

II. HYBRID GENETIC ALGORITHM

Fig.1 shows the flow chart of our HGA. In the step "*Initialization*", we use Quick-Boruvka method for constructing initial tours [9]. The Quick-Boruvka is slightly modified so as to add randomness for producing diversity in the initial population of HGA.

The step "*LK heuristic*" and the step "*Crossover*" are described in detail later. The step "*Selection*" selects individuals according to the ranking of tour lengths. Mutation is not used in this paper.

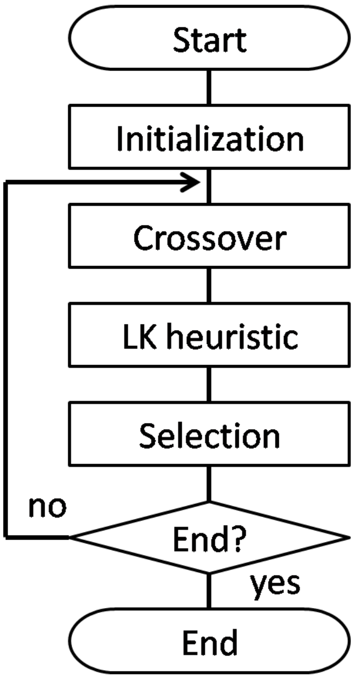

Fig.1. The Flow Chart of HGA

1. Lin-Kernighan Heuristic

Like many other HGA methods for TSPs [10][11], we also use the LK heuristic, because it is widely recognized as powerful local search heuristic for TSPs.

Our LK heuristic uses 8-quadrant nearest neighbors for two-dimensional TSPs, and the search depth is imposed a maximum value of 25. In addition, our LK implementation uses technique of "don't look bits", technique of "distance caching" and data structure of "two-level tree segment list" [6].

2. Chromosome Representation

Tours are represented by so-called *path representation*. For example, a tour "1-2-5-3-4-1" is represented by (12534) by path representation. If a city *i* appears in the *j*th position of the list, city *i* is the *j*th visit.

3. Crossover

The proposed crossover SRX divides each tour of both the parents into many sub-tours that are limited by maximum tour length *K*, and reconnects sub-tours from both the parents so as to construct tour of TSP. The details of the SRX are as follows.

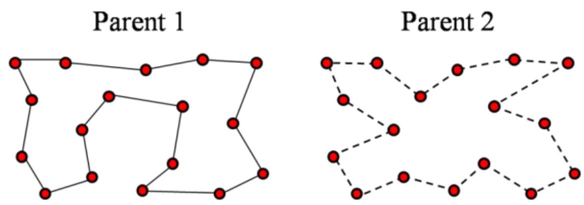

Fig.2. An Example of Parents

Step1

A starting city is selected at random. A sub-tour from parent 1 is extended from the starting city to both directions until the length of the sub-tour is *K*. But if a candidate city to be selected is already copied from the parents, the extension is terminated. (Fig.3)

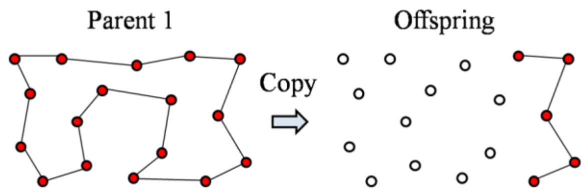

Fig.3. An Example of a Sub-tour Copied from Parent 1 (assume $K = 4$)

Step2

A city is randomly selected as a starting city. A sub-tour from parent 2 is extended to both directions from the starting city until the extension is terminated as Step1. (Fig.4)

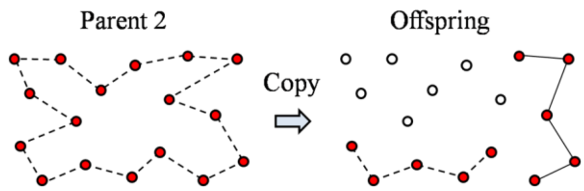

Fig.4. An Example of a Sub-tour Copied from Parent 2 (assume $K = 4$)

Step3

Repeating Step1 and Step2 until all the cities are copied from the parents to offspring. (Fig.5)

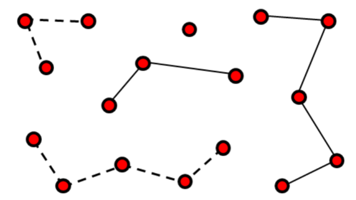

Fig.5. An Example of Sub-tours Copied from both the Parents

Step4

A pair of end cities belonging to different sub-tours are connected so as to makeup a new TSP tour.

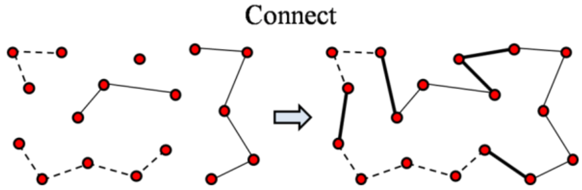

Fig.6. An Example of Offspring Produced by Connecting Sub-tours

III. EXPERIMENTS AND DISCUSSIONS

Benchmark tests evaluate the effectiveness of the SRX. We compare the SRX with the three conventional crossovers; MPX3, GSX2 and ERX6. We used ten benchmarks with sizes from 38 to 71,009 cities taken from the TSP website of Georgia Institute of Technology. Ten time experiments are executed for each benchmark.

We evaluate it from the viewpoint of tour quality and CPU time. The tour quality is measured as gap from the optimum or known best tour if optimum is unknown. In this case study, optima of two instances (bm33708, ch71009) are unknown. *Quality* is defined by the following equation:

$$quality = \frac{tour\_length - optimum}{optimum} \times 100(\%)$$

The program code is written in the C++ language and compiled by GNU g++ compiler (version 2.95) with "-O2" optimizing option. The LK algorithm is programmed by referring CONCORD code [12].

The HGA parameters are set as follows according to preliminary experiments. The population size is 30, maximum generation is 10000, and SRX parameter *K* is $\sqrt{N}$ (N; number of cities).

The HGA is executed on cluster in our Lab. (2.8-GHz Xeon, 6.0 GHz memory, CentOS 5.2).

Experimental results of five larger benchmarks are given in Table 1 and Fig.7. In table 1, number in the instance name is a size of the problem. The bold type shows the best value of each benchmark.

Table 1. Results of Comparison Experiments of Four Crossovers for Large-scale TSPs.

Instance	Crossover	Ave. Quality (%)	Ave. CPU Time (s)
it16862	SRX	**0.173**	**96336.6**
	MPX3	1.235	116145.2
	GSX2	1.566	125236.0
	ERX6	1.117	107793.8
vm22775	SRX	**0.204**	**166895.4**
	MPX3	1.217	227173.2
	GSX2	1.781	238013.4
	ERX6	1.006	236780.6
sw24978	SRX	**0.239**	**155305.0**
	MPX3	1.431	195722.0
	GSX2	1.570	210031.8
	ERX6	1.120	183382.8
bm33708	SRX	**0.274**	**209373.2**
	MPX3	1.320	279107.4
	GSX2	1.632	303155.8
	ERX6	1.162	269690.8
ch71009	SRX	**0.253**	**379655.6**
	MPX3	1.148	604409.4
	GSX2	1.402	723723.0
	ERX6	0.946	682159.6

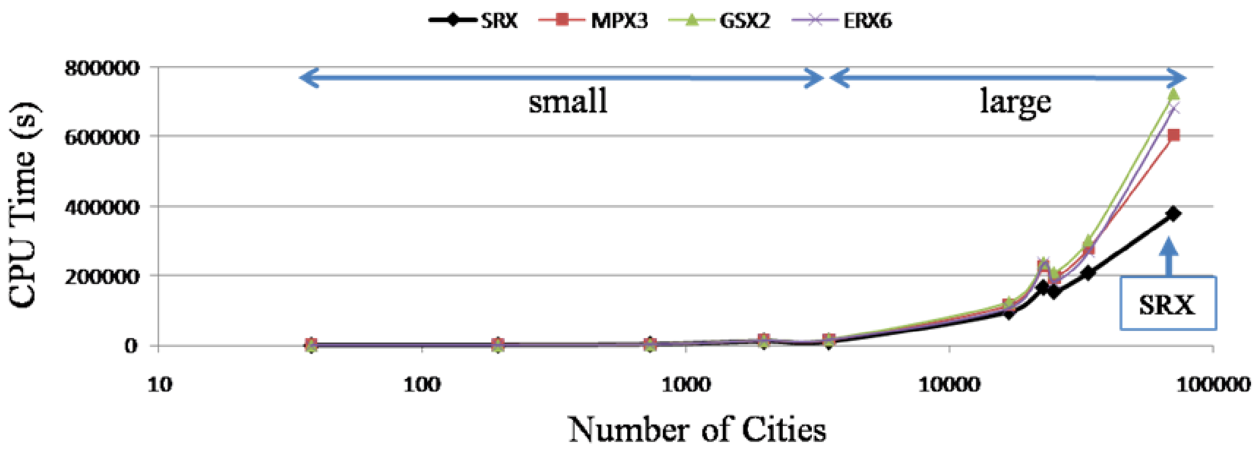

Fig.7. Comparison CPU times of Four Crossovers for Large-scale TSPs

In all benchmarks, the SRX succeeded in finding better solution and running faster than the conventional crossovers.

IV. CONCLUSIONS

We propose a novel crossover operator named SRX of HGA with LK heuristic for large-scale TSPs. Employing ten benchmarks, we compare the SRX to three conventional crossovers (MPX3, GSX2, ERX6) from the viewpoint of tour quality and CPU time. For all benchmarks, the SRX gives best solution compared with other crossovers, and is faster than them.

We intend to parallelize HGA to solve more large-scale TSPs.

ACKNOWLEDGEMENT

This research is partly supported by the collaborative research program 2009, information initiative center, Hokkaido university.

REFERENCES

[1] S. Lin and B. Kernighan (1973), An Effective Heuristic Algorithm for the Traveling Salesman Problem, Operations Research 21(2): 498-516

[2] H. Mühlenbein and M. Gorges-Schleuter and O. Krämer (1988), Evolution Algorithms in Combinatorial Optimization, Parallel Computing, North-Holland 7: 65-85

[3] H. Sengoku and I. Yoshihara (1998), A Fast TSP Solver Using GA on JAVA, Proc. of the 3rd Int. Symp. on Artificial Life and Robotics (AROB-3): 283-288

[4] D. Whitley and T. Starkweather and D. Fuquay (1989), Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator, Proc. Of the 3rd ICGA: 133-140

[5] National TSPs
http://www.tsp.gatech.edu/world/countries.html

[6] H. D. Nguyen (2004), Hybrid Genetic Algorithms for Combinatorial Optimization Problems, Ph. D. dissertation Univ. of Miyazaki

[7] H. D. Nguyen and I. Yoshihara and K. Yamamori and M. Yasunaga (2002), Greedy Genetic Algorithms for Symmetric and Asymmetric TSPs, IPSJ Tech. Report SIG-MPS 43(SIG10): 165-175

[8] H. D. Nguyen and I. Yoshihara and M. Yasunaga (2000), Modified Edge Recombination Operators of Genetic Algorithms for the Traveling Salesman Problem, Proc. Of the 3rd Asia-Pacific Conf. on Simul. Evol. and Learn. SEAL 2000: 2815-2820

[9] D. Applegate and W. Cook and A. Rohe (2003), Chained Lin-Kernighan for Large Traveling Salesman Problems, INFORMS J. of Computing 15(1): 82-92

[10] R. Baraglia and J. I. Hidalgo and R. Perego (2001), A Hybrid Heuristic for the Traveling Salesman Problems, IEEE Trans. EC 5(6): 613-622

[11] N. D. Nguyen and I. Yoshihara and K.Yamamori and M. Yasunaga (2007), Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems, IEEE Trans. SMC 37(1): 92-99

[12] Concorde TSP Solver
http://www.tsp.gatech.edu/concorde/