International Journal of Navigation and Port Research
Vol.29, No.2 pp. 141~146, 2005 (ISSN-1598-7701)

Modified Genetic Operators for the TSP

Sang-Moon Soak · Yeon-Mo Yang** · Hong-Girl Lee*** · Byung-Ha Ahn*****

**, **Department of Mechatronics, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea*
****Department of Logistics Engineering, Korea Maritime University, Busan 606-791, Korea*
*****Prof. Department of Mechatronics, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea*

Abstract : *For a long time, genetic algorithms have been recognized as a new method to solve difficult and complex problems and the performance of genetic algorithms depends on genetic operators, especially crossover operator. Various problems like the traveling salesman problem, the transportation problem or the job shop problem, in logistics engineering can be modeled as a sequencing problem. This paper proposes modified genetic crossover operators to be used at various sequencing problems and uses the traveling salesman problem to be applied to a real world problem like the delivery problem and the vehicle routing problem as a benchmark problem. Because the proposed operators use parental information as well as network information, they could show better efficiency in performance and computation time than conventional operators.*

Key words : *Genetic Algorithms, Sequencing Problems, Genetic Operators, Traveling Salesman Problem*

1. Introduction

Since Genetic Algorithms (GAs) were proposed by John Holland (Holland, 1992), genetic algorithms have been proved as a new method for solving difficult and complex problem for several decades. It is a search algorithm based on the mechanics of natural selection and natural genetics (Holland, 1992). In GAs, a solution is expressed by not just one solution but a population of solutions and then the solutions mate and bear offsprings for the next generation. This reproduction and genetic operators are programmed to replicate the paradigm of survival-of- the-fittest. Over a lot of generations the solutions in the population are improved until the best solution in the population comes close to near optimal.

GAs are consisted of three parts; selection, crossover and mutation. The role of selection is to choose and remain better solutions in a population. The role of crossover operator is to search more precisely near by solution found and the role of mutation operator is to search more widely new search space where it did not explore in the previous generation. So a large number of operators have been developed to improve the performance of GAs because the performance of algorithm depends on the ability of these operators. Especially many researchers have been more interested in crossover operator than other operators because generally a global optimal solution is known as existing near a sub-optimal solution (Boese, 1995).

A broad class of scheduling problems in logistics can be viewed as sequencing problems. By optimizing the sequence of processes or events introduced into a workspace, optimization across the entire problem domain can be achieved. However, a sequencing problem like the traveling salesman problem, the transportation problem or the job shop problem has been used to prove the validity and efficiency of a developed algorithm for a long time. Especially, the traveling salesman problem (TSP) is related to various real world problems like the delivery problem or the vehicle routing problem. Therefore, we use TSP as the test problem for proving the performance of modified genetic crossover operators.

The TSP is easy to describe: given a finite number of "cities" along with the cost (or distance) of travel between each pair of them, find the shortest tour of visiting all the cities and returning to your starting point. If the cost from city A to city B is equal to the cost from city B to city A, the TSP is called a symmetric TSP. If it is not equal, the TSP is called a asymmetric TSP. In this paper, we deal with symmetric TSP.

The first papers tried to solve the TSP using genetic algorithm (GA) were the study of Grefensetette et. al (Grefenstette et. al, 1985) and Goldberg et. al (Goldberg et.

* Corresponding Author : Sang-Moon Soak, soakbong@gist.ac.kr, 062)970-2426
** yangym@gist.ac.kr, 062)970-2426
*** leehg@hhu.ac.kr, 051)410-4911
**** bayhay@gist.ac.kr, 062-970-2402

al, 1985). Since that, many researchers had made many efforts for developing new genetic algorithms, especially crossover operators. Nevertheless, just permutation-based crossover operators such as Cycle crossover (CX) (Oliver et. al, 1989), Order Crossover (OX) (Davis, 1985), Position-Based Crossover (PBX) (Syswerda, 1991), Order-Based Crossover (OBX) (Syswerda, 1991) and Partial-Mapped Crossover (PMX) (Goldberg et. al, 1985) were the main stream. But recently researcher's interest has been changed more and more in the direction of using information of parents (Grefenstette et. al, 1985: Whitley et. al, 1989; Starkweather et. al. 1991). Because the basic mechanism of GAs is to get better solution through information propagated for generation to generation, it seems to be a natural trend. However, the operators using parental information did not also give enough good performance. But, we have an intuition that using the parental information can give a good performance. So, we introduce new operators.

In this paper, we propose modified genetic crossover operators, Edge Preservation Crossover (EPX) and Simple Edge Preservation Crossover (SEPX) focused on improving the performance of genetic algorithms in both performance and speed and compare to well known general operators and edge-based operators. When generating an offspring, the proposed operators can utilize edge information of parents to select the best one among the candidates. Because the proposed operators can propagate the favorable features existing in the selected parents over a lot of generations and reproduce offsprings from the information, they can guarantee that the offsprings are better than their parents with high frequency.

The rest of the paper is organized as follows. The next section introduces several genetic crossover operators. Section 3 gives an outline of the proposed operators and our algorithm. Section 4 presents the results of the experiments. The final section contains some concluding remarks.

2. Genetic Crossover Operator

Five permutation-based crossover operators (PMX, OX, OBX, PBX, CX), heuristic crossover operator and a variant of heuristic crossover operator and two edge-based crossover operators (edge recombination crossover (ER) operator and enhanced edge recombination crossover (EER) operator), are compared to our new crossover operators.

The Permutation-based crossover operators appear more often in the literature. The main advantages of those are that they are very clear and obvious and thus they are easier to implement than other operators and faster than operators using parental information. These kinds of crossover operators can be viewed as variants of conventional crossover operators (one, two or multi-point crossover). Thus, they use kinds of repairing procedures to resolve the illegitimacy of offspring involved by conventional crossover operators in case of TSP.

However, we describe only heuristic crossover operator and its variant, and two edge-based crossover.

2.1 Heuristic Crossover and its variant

The heuristic crossover (HX) was proposed by Grefensetette et. al (1985). It was the first crossover operator using adjacency information of parents. Since that, several variants had been proposed.

Fig. 1 shows the operation process of heuristic crossover. The shaded rectangular presents cities randomly selected. First, heuristic crossover generates offsprings as follows.

Two parents are selected and picks randomly a city (city 6) in two parents. And then choose the shorter edge between next two right side cities (E_{69} and E_{65}) and connect from the current city if it does not make a cycle (in this problem, city 5 is selected). If the shorter edge would make a cycle, it chooses a random city among cities which were not selected at previous steps (in case of city 4 and city 8). This process continues until a tour is completed.

However, HX only uses right side neighbors among parental cities. The symmetric TSP, E_{ij} and E_{ji} are totally the same. Therefore, we think that there is no reason to check only right side edges of parents. So we will expand it checking both side edges.

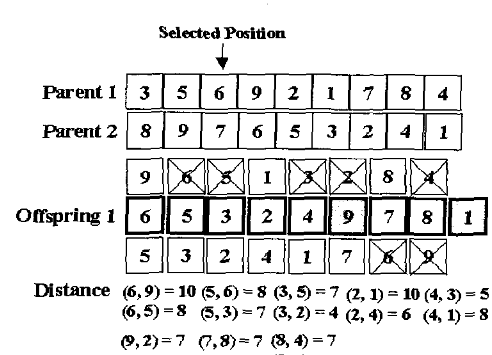

Fig. 1 Heuristic Crossover

One of the variants of heuristic crossover is greedy crossover (GX) (Yang, 1997). The difference between HX and GX is that in GX, offspring1 is generated by the right side edges of parents and offspring 2 is generated by the left side edges of parents.

2.2 Edge Recombination Crossover and its variant

The edge recombination crossover (ER) was proposed by Whitley et. al (1989) for solving TSP. It utilizes only adjacent information of nodes in parents instead of distance information of parental edges. It first makes an adjacent table of each node in parents and uses the table for generating offsprings. Fig. 2 shows the edge recombination crossover. First, it chooses randomly a city (city 3) and among the elements that have a link with current city, chooses the element which has the fewest number of links remaining in its adjacent table entry (in this problem, city 4 has four elements and city 5 has 2 element. So city 5 is selected). If elements have the same number of links (in case of city 6), choose randomly one (city 9) among them (city 7 and city 9). This process continues until it generates a complete tour.

The modified version of ER is enhanced edge recombination crossover (EER) proposed by Starkweather et al (Starkweather et. al. 1991). The difference between the original version and the variant is that the variant preserves the common edges of parents.

The EER has a new adjacent table that is the same as the old adjacent table, except for the tagged information of some common edges between the parents. That is, common edge between parents puts a tag and gives the tagged edge to a priority at competition.

They showed that ER and EER using adjacency information of parents are to get better solution than that of other conventional crossover operators (Whitley et. al, 1989; Starkweather et. al. 1991). After EER, more variants had been introduced in the literature for enhancing the original version (Mathias and Whitley, 1992; Nguyen et. el, 2000).

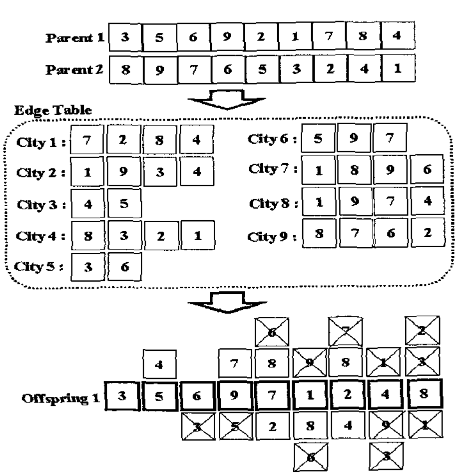

Fig. 2 Edge Recombination Crossover

3. EPX Operator

New crossover operators were developed by studying HX (Grefenstette et. al, 1985) and EER (Starkweather et. al. 1991). The former is focused on the distance of edges, whereas the latter gives the priority at common edge. We can not assert what more important information between distance of edges and common edge of parents is. So we have developed two crossover operators; one is an operator considering only distance of edges, the other is an operator considering distance of edges but giving priority at common edge of parents. But both the two operators are different from the previous two operators. The proposed crossover operator can utilize much more parental information.

First, the simple edge preservation crossover (SEPX) is focused on distance of edges. But the difference with HX is that HX checks only right side edges of current city, on the contrary SEPX checks not only the right side but also left side edges of current city. As mentioned in the previous section, there is no reason to check only right side edges because edge (a, b) and edge (b, a) present the same information in case of symmetric TSP.

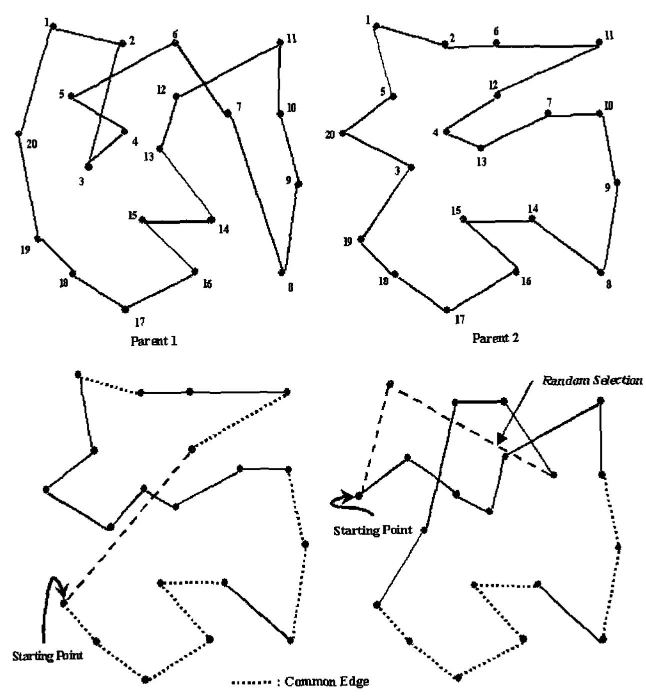

Fig. 3 An example of EPX

Next one is the edge preservation crossover (EPX). EPX combines the idea on distance of edges and on giving priority at common edge. We assume that common edges may include more important information augmented for evolutionary process. Fig. 3 shows an example of EPX and the procedure is as follows. In here, we explain the procedure focusing on offspring 1.

One arbitrary city (city 19) is chosen and four edges ((19, 20), (19, 18), (19, 18), (19, 3)) of current city in parents are checked. If a common edge exists, the common edge is selected as next city ((19, 18) is the common edge). Otherwise, compare with distances of each edge and the shortest edge are selected. If the shortest one is already selected at the previous step, the next shorter edge is considered. If all edges were introduced at a previous process, a city which is not selected at the previous steps is selected randomly.

The advantage of SEPX and EPX is to generate offsprings as many as we want. That is, if we select different starting points, it can generate different offsprings. Fig. 3 shows two parents selected for crossover and offsprings generated by EPX. In this paper, number of offspring ($N_{Offspring}$) is fixed as 2 for fair comparison with other crossover operators.

4. Experimental Results

In this paper, we perform two comparative experiments. One compares several crossover operators and the other analyzes the stability of EPX and SEPX. In the first experiment, for comparison with each crossover operator, we use a simple GA, which does not include the local search method. But in the second experiment, we use a local search method as well as the proposed crossover operators. The used local search method is 2-opt algorithm. This algorithm deletes two edges, thus breaking the tour into two paths, and then reconnects those paths in a possible way.

First, we describe the framework of the GA in the experiment. Initial individuals are randomly generated and used binary tournament strategy for the selection. The elitist selection strategy is also applied. That is, if the best tour in the new generation is worse than the previous one, we preserve the previous best tour by passing it to the new generation. The number of preserved best tour is 3. The mutation operator used in this experiment is the inversion mutation operator. It first selects arbitrarily two positions in a chromosome and then inverts the substring between these positions.

The population size (*pop_size*), the crossover probability (P_c) and mutation probability (P_m) are 200, 0.6 and 0.4 respectively. The termination condition is that if the best tour so far has not been improved for more than N generations, the whole algorithm is stoped. In here, N is currently set to 1,000. We carry out 30runs on each operator by changing the seed of the random number.

All algorithms are implemented by using Visual C++ on PentiumIII 450Mhz. For comparing with operators, the four test instances (eil51, eil76, kroA100 and pr124) are used and for confirming the abilities to find the best known solution, another four test instances (lin105, pr107, pr124 and kroA150) are used. They are all from the TSPLIB [14]. In all experiment, the quality (minimum, maximum and average) is defined as follows, i.e. the quality indicates the percentage over the optimal value.

$$quality = \frac{Fitness - Optimal}{Optimal} \times 100(\%) \quad (1)$$

Table 1 shows the result of first experiment. EPX and SEPX found better solution than other crossover operators at most of the cases. In the case of eil51 instance, the proposed crossover operators found the optimal solution, but the other crossover operators did not find optimal solution. In the other instances, the proposed operators did not find optimal solution but found near optimal solution. The solution quality did not exceed 3.9% (EPX) and 2.5% (SEPX) respectively at pr124. EPX and SEPX required a little more computation time than permutation-based operators. On the other hand, they spent much less computation time than operators (ER and EER) using adjacency information of parents.

In the comparison between EPX and SEPX, SEPX showed better results than EPX but needed more computation time and more generation. We could confirm that the proposed crossover operators are superior to permutation -based crossover operators and edge-based crossover operators.

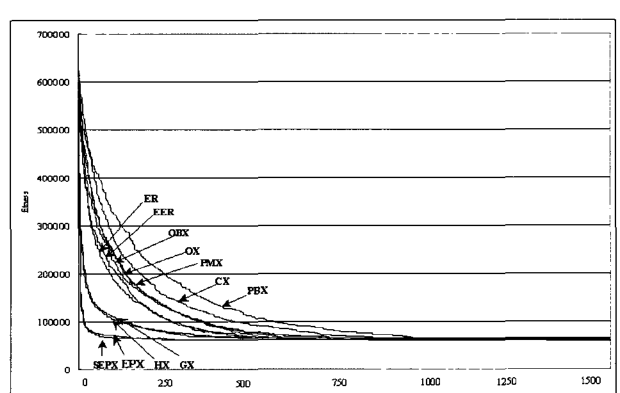

Fig. 4 The results of pr124 problem

Fig. 4 shows the results at pr124 instance. In this figure, it is an interesting fact that operators using adjacent

Table 1 The comparison of crossover operators

	eil51 (426)					eil76 (538)				
	Min	Avg.	Max	Avg. CPU Time(sec)	Avg. gen	Min	Avg.	Max	Avg. CPU Time(sec)	Avg. gen
PMX	434	448.8	463	11.6	1614.2	568	582.8	605	18.0	1521.7
OX	431	444.6	457	16.3	1916.1	545	560.6	579	26.2	2014.9
OBX	431	445.3	463	17.9	1848.9	556	572.5	589	41.3	2420.8
PBX	431	446.8	468	24.4	1888.9	556	575.3	596	56.3	2435.8
CX	435	448.7	463	8.8	1540.5	552	583.5	608	13.9	1638.1
ER	435	445.9	462	35.8	1762.9	561	578.2	609	73.7	2048.8
EER	435	447.5	465	28.8	1448.8	561	580.8	604	61.7	1734.7
HX	427	438.9	450	25.2	1577.2	552	566	586	52.1	1895.6
GX	435	447	467	19.6	1308.6	551	580.5	599	37.3	1434.8
EPX	426	435.4	450	17.4	1080.4	544	561.8	577	36.1	1235.9
SEPX	426	434.2	450	28.6	1365.8	541	552.5	563	48.7	1553.6
	kroA100 (21282)					pr124 (59030)				
	Min	Avg.	Max	Avg. CPU Time(sec)	Avg. gen	Min	Avg.	Max	Avg. CPU Time(sec)	Avg. gen
PMX	22165	23159	24782	29.9	1771.4	60362	62654.9	66679	48.6	2230.8
OX	21389	22712	24045	42.6	2071.6	59777	62048.7	64401	63.2	2300.1
OBX	21876	22911	24341	67.3	2581.5	59576	61260.7	65125	80.6	2869.0
PBX	21543	23280	24503	90.2	2545.4	60163	62245	65693	137	2780.9
CX	21959	23003.8	24854	21.4	1943.5	59652	62061.3	64837	29.6	2389.6
ER	21579	22882	24383	102.3	1849.2	60049	62265.5	70275	163.2	2102.4
EER	21941	23031.3	24185	92.3	1672.6	59903	62104.5	66618	155.7	2023.6
HX	21644	22504	23848	70.7	1704.5	59413	61147.8	63700	116.5	2007.7
GX	22261	23029.1	24255	62.9	1615.9	60560	62362.2	67275	104.4	1910.9
EPX	21556	22263.9	24009	59.7	1392.8	59087	61359.5	63906	90.6	1521.1
SEPX	21383	21894.4	22798	75.3	1530.8	59323	60561.5	63297	104.3	1538.7

( ) : the optimum solution.

information present more early convergence and find better solution. It shows a possibility to find much better solution in much less computation time if the proposed operators are combined with a local search method.

Next experiment is the stability analysis of EPX and SEPX operator. In this experiment, we analyze the stability of the proposed crossover operators because it is another important measure.

As mentioned above, 2-opt is used for an extensive local search. Although it needs an amount of computation time, it can find better solution.

The results of EPX and SEPX at lin105, pr107, pr124 and kroA150 instances are shown in Table 2. CPU Time shows the average CPU Time until satisfying the termination condition. In here, EPX and SEPX found a known optimal solution in all test cases and obtained average quality of nearly 0 %. In all of the cases, SEPX generated better quality than EPX and required much less computation time because SEPX found the optimal solution in a few generation. Although we can not assert that the distance of edges is much important information than common edges of parents, apparently it seems to be that using the former can give a slightly better results that using the latter.

5. Conclusion

In this paper, we have experimented new crossover operators, simple edge preservation crossover (SEPX) and edge preservation crossover (EPX) for sequencing problems.

By means of various experiments, it has been confirmed that SEPX and EPX can get better results than permutation-based crossovers and operators using parental information. Although the proposed new crossover operators needed more computation time than that of permutation-based crossovers, the performance is much superior to them. And we could confirm that using parental information can be a technique for finding good solutions. The stability analysis of EPX and SEPX showed that SEPX is better than EPX in all aspects of experimental. But we can not prematurely convince that distance information between cities is more important than common edge information. And the proposed crossover operators also can be applied to other sequencing problems if a fitness function can be properly changed based on the global information for a given problem instead of the distance for the traveling salesman problem.

There is an interesting direction for a future work. Both

Table 2 The stability analysis of EPX & SEPX

	Quality(%)											
	lin105						pr107(44303)					
	Min	Avg.	Max	Avg. CPU Time(sec)	Best	Avg. gen	Min	Avg.	Max	Avg. CPU Time(sec)	Best	Avg. gen
EPX	0.0	0.0	0.0	30.8	30/30	109.8	0.0	0.057	0.305	145.4	24/30	515.5
SEPX	0.0	0.0	0.0	25.7	30/30	87.2	0.0	0.049	0.494	110.7	25/30	380.8
	pr124(59030)						kroA150(26524)					
	Min	Avg.	Max	Avg. CPU Time(sec)	Best	Avg. gen	Min	Avg.	Max	Avg. CPU Time(sec)	Best	Avg. gen
EPX	0.0	0.008	0.078	48.1	27/30	126.9	0.0	0.55	1.78	958.1	1/30	1688
SEPX	0.0	0.005	0.078	37.0	28/30	94.9	0.0	0.56	2.0	871.9	3/30	1499

( ) : the optimum solution.

of the proposed operators take more computation time than the others. So, we will investigate how to reduce the computation time. And the proposed operators may not preserve a common sub-tour in a complete tour of two parents. So it can lose important information of parents. This problems will be investigated to further improve the proposed operators.

Reference

[1] Boese, K.D.(1995), "Cost Versus Distance In the Traveling Salesman Problem", Technical Report CSD-950018, UCLA Computer Science Department.

[2] Davis, L.(1985) "Applying Adaptive Algorithms to Domains" Proc. of the International Joint Conference on Artificial Intelligence, pp. 162~164.

[3] Goldberg, D. and Lingle, R.(1985) "Alleles, Loci and the Traveling Salesman Problem", Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 154~159.

[4] Goldberg, D.(1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley.

[5] Grefenstette, J. Gopal, R. Rosmaita, B. and Gucht, D.(1985) "Genetic Algorithms for the Traveling Salesman Problem", Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp.160~168.

[6] Holland, J.H.(1992), Adaptation in Natural and Artificial Systems, The MIT Press.

[7] Mathias, K. and Whitley, D.(1992) "Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem", Parallel Problem Solving from Nature 2, pp.219~228, North Holland-Elsevier.

[8] Nguyen, H. D. Yoshihara, I. and Yasunaga, M.(2000) "Modified Edge Recombination Operators of Genetic Algorithms for the Traveling Salesman Problem", Industrial Electronics Society, IECON 2000. 20 th Annual Conference of the IEEE, Vol. 4, pp. 2815~2820.

[9] Oliver, I. Smith, D. and Holland, J.(1989) "A Study of Permutation Crossover Operators on the Traveling Salesman Problem", Proc. of the Second International Conference on Genetic Algorithms, pp. 224~230, July.

[10] Starkweather, T. McDaniel, S. Mathias, K. Whitley, D. and Whitley, C.(1991) "A Comparison of Genetic Sequencing Operators", Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, CA.

[11] Syswerda, G.(1991) "Schedule Optimization Using Genetic Algorithms", In L. Davis, ed., Handbook of Genetic Algorithms, pp. 332~349.

[12] TSPLIB. Web Site, http://www.iwr.uni-heidelberg.de /iwr/comopt/soft/TSPLIB95/TSPLIB.html.

[13] Whitley, D. Starkweather T. and Fuquay, D.(1989) "Scheduling Problems and Traveling Salesman: the Genetic Edge Recombination and Operator", Proc. Third Int. Conf. Genetic Algorithms and their Applications, pp.133~140.

[14] Yang, R.(1997) "Solving Large Traveling Salesman Problems with Small Populations" Genetic Algorithms in Engineering Systems: Innovations and Applications, pp.157~162.

Received 27 January 2005
Accepted 11 March 2005