Effect of Local Search on Edge Histogram Based Sampling Algorithms for Permutation Problems

Shigeyoshi Tsutsui[*] Martin Pelikan[†] Ashish Ghosh[$]

[*] Dept. of Management and Information Science, Hannan University,
5-4-33, Amamihigashi, Matsubara, Osaka 580-8502 Japan. tsutsui@hannan-u.ac.jp

[†] Dept. of Math and Computer Science, University of Missouri at St. Louis
8001 Natural Bridge Rd., St. Louis, MO 63121. pelikan@illigal.ge.uiuc.edu

[$] Machine Intelligence Unit, Indian Statistical Institute
203 B. T. Road, Kolkata, 700 108, India. ash@isical.ac.in

1 Introduction

One of the most promising research directions that focus on eliminating the drawbacks of fixed, problem-independent genetic algorithms, is to look at the generation of new candidate solutions as a learning problem, and use a probabilistic model of selected solutions to generate the new ones [5,9,10]. The algorithms based on learning and sampling a probabilistic model of promising solutions to generate new candidate solutions are called *probabilistic model-building genetic algorithms (PMBGAs)* [9,10], *estimation of distribution algorithms (EDAs)* [7], or *iterated density estimation algorithms (IDEAs)* [1].

Despite the great success of PMBGAs in the domain of fixed-length discrete and continuous vectors, only few studies are found for permutation and scheduling problems [1]. More importantly, these studies take an indirect approach of mapping the permutation problems to fixed-length vectors of discrete or continuous variables, which in some cases necessitates the use of repair operators to correct invalid solutions. Tsutsui [12] introduced a promising approach to use PMBGAs for permutation problems using *edge histogram based sampling algorithms (EHBSA)*, and showed competitive results on several benchmark instances of the traveling salesman problem (TSP) and flow shop scheduling problem [13].

In this paper EHBSA has been hybridized with some popular local search heuristics to take advantage of local search. We tested two types of heuristics; one is a simple heuristic for solving TSP, called 2-OPT; and the other is a sophisticated one, known as Lin-Kernighan heuristic [4,6]. Experimental evidences show that EHBSA added with these heuristics can solve TSPs even with thousand of cities (requiring less computation) fairly well; providing a method capable of solving significantly larger problems. The method is very robust in the sense of detecting global optima for a wide variety of population sizes. On the other hand, a combination of the local heuristics with existing recombination (e.g., OX, PMX, eER) based techniques also show advantage, but not so reliable and robust. This is due to the fact that they can detect the global optimum mostly for a bigger population size.

2 Edge Histogram Based Sampling Algorithms: A Brief Overview

Here we brief EHBSAs, and their use for (i) modeling promising solutions, and (ii) generating new solutions. A set of edges (including all nodes) that generates a path in a graph can be used as an alternative representation of a permutation of the nodes in the graph. The basic idea of EHBSA is to collect and exploit information about the presence of edges (frequency of edges) in the entire population of selected high-quality solutions.

2.1 Edge Histogram Matrix

Let the permutation represented by the k-th individual in a population $P(t)$ of size N, at generation t, be denoted by $s^t_k = (\pi^t_k(0), \pi^t_k(1), ..., \pi^t_k(L\text{-}1))$, where $(\pi^t_k(0), \pi^t_k(1), ...,\pi^t_k(L\text{-}1))$ define a permutation of $(0, 1, ..., L\text{-}1)$, and L is the length of the permutation. The edge histogram matrix EHM^t $(e^t_{i,j})$ $(i, j = 0,1, .., L\text{-}1)$ of the population is symmetrical and consists of L^2 elements. The (i,j)-th element of the matrix is obtained by

$$e^t_{i,j} = \begin{cases} \sum_{k=1}^{N} (\delta_{i,j}(s^t_k) + \delta_{j,i}(s^t_k)) + \varepsilon & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}, \tag{1}$$

where $\delta_{i,j}(s^t_k)$ is a delta function defined as

$$\delta_{i,j}(s^t_k) = \begin{cases} 1 & \text{if } \quad \exists h\,[h \in \{0,1,\cdots L-1\} \wedge \pi^t_k(h) = i \wedge \pi^t_k((h+1) \bmod L) = j] \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

and ε(> 0) biases the sampling toward random permutations. To have similar selection pressure, ε should be proportional to the expected value of $e^t_{i,j}$. Since the average value of $e^t_{i,j}$ for $i \neq j$ in EHM^t is $2LN/(L^2\text{-}L) = 2N/(L-1)$, we use

$$\varepsilon = \frac{2N}{L-1} B_{\text{ratio}} \tag{3}$$

where B_{ratio} (> 0) is a constant related to the pressure toward random permutations. A smaller value of B_{ratio} reflects the real distribution of edges in the parent population.

2.2 Sampling Algorithms

Two algorithms, namely, edge histogram based sampling algorithm without template (EHBSA/WO) & edge histogram based sampling algorithm with template (EHBSA/WT) were used for sampling the EHM^t.

2.2.1 EHBSA/WO

Let us denote the elements of the permutation, to be sampled, by $c[i]$ for $i \in \{0, 1, \ldots, L\text{-}1\}$. EHBSA/WO starts by randomly selecting the initial element of it, denoted by $c[0]$. Elements are selected using the roulette-wheel selection algorithm [3]. Let us assume that the last element generated is $c[i]$. The new element $c[i+1]$ is set to j (we restrict potential values of j so that $j \neq c[k]$ for all $k \in \{0, 1, \ldots, i\}$; to avoid creating cycles) with a probability proportional to $e^t_{c[i],j}$ of EHM^t. This continues until the entire permutation has been generated. Figure 1 shows the schematic description of EHBSA/WO.

1. Set the position counter $p \leftarrow 0$.
2. Obtain first node c[p] randomly from $\{0, 1,2,\ldots.L\text{–}1\}$.
3. Construct a roulette wheel vector $rw[]$ from EFM^t as $rw[j] \leftarrow e^t_{q\ j,}$ (j=0, 1, .., L–1).
4. Set 0 to previously sampled nodes present in $rw[]$ ($rw[\ c[i]\ ] \leftarrow 0$ for i =0, .., p).
5. Sample next node $c[p+1]$ with probability $rw[x] / \sum_{j=0}^{L-1} rw[j]$ using roulette wheel $rw[]$.
6. Update the position counter $p \leftarrow p+1$.
7. If $p<L-1$, go to Step 3 else stop.

Figure 1: EHBSA/WO

2.2.2 EHBSA/WT

EHM^t gives a marginal edge histogram and has no graphical structure. EHBSA/WT is intended to make up for this disadvantage by using a template in sampling new individuals. For generating a new individual, a template individual is chosen from $P(t)$ (randomly). Using n (n>1) random cut points the template is divided into n+1 segments. We choose one segment randomly and sample nodes for that segment. Nodes for other segments remain unchanged. We denote this sampling method by EHBSA/WT/n.

All evolutionary models are based on the steady-state scheme. If the offspring is better than the (worst) parent, we incorporate it into the population.

2.4 Results and Analysis

Here we show results of 20 simulations for all problems and parameter settings. Each run was terminated either when the optimal tour was found, or when the population converged, or when the number of evaluations reached $E_{\max}$ (supplied by the user). Values of $E_{\max}$ were 500000 and 1000000 for pr76. Population size 60, 120, 240 were used for EHBSA; for other algorithms, population size were 240, 480 and 960. B_{ratio} of Equation 3 is set to 0.005 for all experiments.

We evaluated the performance by measuring: *#OPT* (the number of runs in which the algorithm succeeded in finding the optimal tour), *MNE* (the mean number of evaluations to find the global optimum in those runs where it did find the optimum), and *Error* ((the average length of the best solution over the 20 runs – optimal tour length) / optimal tour length).

Results on pr76 problem are shown in Table 1. EHBSA/WO could not find the optimum tour, whereas EHBSA/WT/n found the optimal tour in almost all cases. With N = 60, EHBSA/WT/2, 3, 4, and 5 found the optimal tour 13, 14, 14, and 18 times, respectively. With N = 120, EHBSA/WT/2, 3, 4, and 5 found the optimal tour 19, 19, 18, and 19 times, respectively. With

Table 1: Results of pr76

Pop size		60				120				240			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
EHBSA	WO	0/20	-	-	0.0560	0/20	-	-	0.0288	0/20	-	-	0.0764
	WT/2	13/20	339947.6	60683.5	0.0021	19/20	615354.5	88813.6	0.0001	19/20	1083794.9	154414.8	0.0004
	WT/3	14/20	395428.8	166260.6	0.0009	19/20	712166.7	195965.6	0.0001	20/20	1117316.7	131584.2	0
	WT/4	14/20	503356.7	154634.0	0.0017	18/20	866578.9	197105.3	0.0008	18/20	1595085.9	215993.2	0.0001
	WT/5	18/20	707941.4	200621.3	0.0006	19/20	1375098.1	237743.3	0.0001	6/20	1812298.3	201695.2	0.0016
Pop size		240				480				960			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
OX		0/20	-	-	0.1575	0/20	-	-	0.1112	0/20	-	-	0.0766
PMX		0/20	-	-	1.2795	0/20	-	-	0.9500	0/20	-	-	0.6934
EER		0/20	-	-	0.0725	1/20	109242.0	0.0	0.0231	2/20	261935.0	6349.0	0.0076

N=240 EHBSA/WT had similar values of #OPT, and larger value of MNE. Thus, we can see that the performance of EHBSA/WT is much better than EHBSA/WO. Among the other operators, only eER [11] was able to find the optimal tour in one run with N = 480 and in 2 runs for N = 960. OX [8] and PMX [3] could not find the optimal tour at all.

3 Improving Performance of EHBSA with Local Search

In this section we examine a hybrid approach, where EHBSA is combined with local search heuristics for improving solutions. We tested two types of heuristics; one is a simple heuristic for solving TSP called 2-OPT, and the other is a sophisticated Lin-Kernighan [4,6] heuristic.

3.1 2-OPT Heuristic

2-OPT proceeds by checking pairs of nonadjacent edges in a given tour, and computing the improvement in the tour length after rearranging these pairs by exchanging the terminal nodes of the two edges in each pair as shown in the adjacent figure. If no pair of edges can be rearranged to improve the current tour, the algorithm is terminated. Otherwise, the pair of edges that improves the performance maximum (minimum tour length) is rearranged, and the algorithm is executed again.

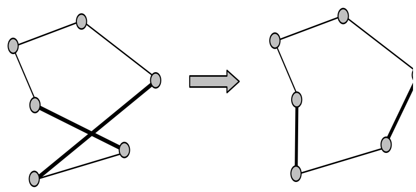

Figure 2: One step of 2-OPT

Results of EHBSA with 2-OPT

For these experiments the maximum number of evaluations is set to 100,000. Other settings are the same as described in Section 2. Table 2 shows the results on the 318-city problem lin318. EHBSA/WO showed lesser value of MNE than EHBSA/WT. However, its success rate is lower than EHBSA/WT. For this problem, EHBSA/WT/3 with population size N=30 showed #OPT=20 and MNE=11928.4. Although #OPT increased with n of EHBSA/WT/n, values of MNE also increased. This confirms the advantages of using local search in combination with EHBSA (EHBSA/WT is better than EHBSA/WO). The performance of OX and PMX are mostly worse than EHBSA, but for some settings these operators yielded comparable performance. Extended ER performed the worst.

Table 3 shows that increasing the number of cities from 318 to 439 necessitated a slight increase in the population size needed by EHBSA. For many settings, MNEs for the 439-city problem decreased compared to the smaller problem of 318-cities presented in Table 2, which indicates good scalability of EHBSA. However, other recombination operators show a significant decrease in performance compared to the 318-city problem. Extended ER is not capable of achieving reliable convergence to the optimum even with a population size N=960, whereas convergence of PMX and OX became reliable only with N=960, showing very inferior performance compared to EHBSA/WT with almost all settings.

Table 2: Results for lin318

Pop size		15				30				60			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
EHBSA	WO	2/20	2305.5	108.5	0.0019	12/20	5580.3	976.4	0.0005	18/20	11664.1	1613.2	0.0002
	WT/2	10/20	7830.7	9183.7	0.0003	18/20	7850.9	2097.9	0.0001	20/20	15220.1	2706.8	0
	WT/3	16/20	8993.6	6306.7	0.0003	20/20	11928.4	6790.4	0	20/20	19997.8	6466.6	0
	WT/4	20/20	15970.7	10235.3	0	20/20	20865.1	9506.9	0	20/20	30609.1	12317.4	0
	WT/5	20/20	22430.1	19460.7	0	20/20	25825.0	13329.0	0	20/20	35579.2	16460.6	0
Pop size		240				480				960			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
OX		5/20	8660.6	1815.8	0.0009	15/20	16291.3	3390.2	0.0003	20/20	26704.1	3297.2	0
PMX		6/20	7413.5	1675.6	0.0008	13/20	14219.2	2521.2	0.0003	20/20	29290.9	3683.7	0
EER		3/20	10815.3	372.0	0.0006	12/20	21156.8	1375.3	0.0002	17/20	42207.2	5354.5	0.0001

Table 3: Results for pr 439

Pop size		15				30				60			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
EHBSA	WO	0/20	NaN	NaN	0.0008	0/20	NaN	NaN	0.0006	7/20	9234.7	1827.4	0.0002
	WT/2	6/20	2681.8	917.6	0.0003	13/20	5659.3	2214.6	0.0001	18/20	10284.9	2107.9	0.0000
	WT/3	6/20	11649.8	13536.9	0.0002	15/20	9084.7	3920.3	0.0001	19/20	14321.3	3727.9	0
	WT/4	14/20	8812.1	5820.7	0.0001	19/20	13021.1	6705.4	0.0000	20/20	17126.9	3509.9	0
	WT/5	11/20	19993.1	18286.6	0.0001	19/20	24371.4	17549.8	0.0000	20/20	25006.0	5944.2	0
Pop size		240				480				960			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
OX		1/20	49358.0	0.0	0.0007	4/20	15203.5	999.0	0.0004	10/20	29125.3	3199.9	0.0001
PMX		2/20	8532.0	425.0	0.0008	5/20	17100.0	2470.6	0.0002	10/20	31571.7	4176.3	0.0002
EER		1/20	10815.3	0.0	0.0003	4/20	25714.8	1980.0	0.0001	8/20	65775.5	20199.8	0.0001

Summarizing the results of hybridization with 2-OPT, we can say that EHBSA provided robust performance over all population sizes; whereas other methods provided much inferior output with respect to reliability and computational efficiency. #OPT found by EHBSA was always more than those by other operators (even with a larger population size). Correspondingly, MNE is much less for EHBSA than those of other operators. Comparing EBHSA/WO and EBHSA/WT we can say that EBHSA/WT mostly shows superior performance.

3.2 Lin-Kerninghan heuristic

Lin-Kernighan (LK) [4,6] local search is based on 2-OPT moves. The current solution is viewed as an *anchored* Hamiltonian path P rather than as a Hamiltonian circuit. The *anchor* of the path is a fixed end-point city t_1, as illustrated in the first part of Figure 3. Let t_{2i} denote the other endpoint of the path P_i that exists at the beginning of step i of the LK search. The tour corresponding to path

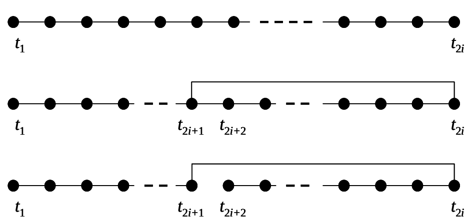

Figure 3: One step of LK

P_i can then be obtained by adding the edge $\{t_{2i},t_1\}$. At each step we only consider 2-Opt moves that flip some suffix of the path, i.e. one of the tour edges being broken is $\{t_1,t_{2i}\}$. Furthermore, the new neighbor t_{2i+1} of t_{2i} must be such that the length of the *one-tree* (spanning tree plus one edge) obtained by adding the edge $\{t_{2i},t_{2i+1}\}$ to P_i (middle part of Figure 3) is less than the length of the best tour seen so far. This restriction is a generalization of the criterion in 2-Opt that $d(t_2,t_3)$ be less than $d(t_1,t_2)$. There are many implementations available for LK search. Here we used an iterative implementation of Concord TSP Solver available at http://www.tsp.gatech.edu/concorde.html.

Results of EHBSA with Lin-Kernighan heuristic

Results for a 3795-city problem fl3795 using EHBSA sampling combined with LK local search are given in Table 4, and those for a 5934-city problem rl5934 in Table 5. For these experiments the maximum number of evaluations is set to 1,000 and 3,000 for fl3795 and rl5934, respectively. The remaining settings are the same as those described in Section 2.

It is evident from the results given in Tables 4 & 5, that incorporation of LK search heuristics with either the recombination based standard algorithms or EHBSA/WT improves (in terms of both MNE and #OPT) the performance. They can solve bigger problems even with 5934 cities quite well. This establishes the utility of LK search in combination with evolutionary algorithms. It is seen from the Tables that for all population sizes (even for a small population size of 5) EHBSA/WT combined with a LK heuristic succeeded in finding out the optimum solution for all runs. But other operators (OX, PMX, eER) combined with LK heuristics could not find the optimum in all runs for smaller population sizes. Let us take the case of population size 15 and pr5934 problem. The EHBSA/WT could detect the optimum 100% times; whereas the OX did it for 11 times, PMX for 15 and eER for 17 times out of 20. They could succeed in all runs only when the population size was as big as 60. Other measures like MNE, STD are comparable for all techniques. Similar are the results even for the 3795

Table 4: Results for fl3795

Pop size		5				10				15			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
EHBSA	WT/2	20/20	55.6	24.1	0	20/20	81.1	35.3	0	20/20	78.5	40.7	0
	WT/3	20/20	70.4	69.5	0	20/20	76.7	35.9	0	20/20	60.4	32.7	0
	WT/4	20/20	94.3	64.9	0	20/20	67.0	36.0	0	20/20	65.4	34.7	0
	WT/5	20/20	81.6	79.4	0	20/20	79.1	52.1	0	20/20	86.0	40.6	0
Pop size		10				15				30			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
OX		17/20	43.4	14.4	0.0003	19/20	49.8	23.9	0.0001	20/20	77.9	31.4	0
PMX		18/20	50.7	22.5	0.0008	19/20	60.1	30.2	0.0001	20/20	84.8	27.8	0
EER		20/20	149.1	70.5	0.0000	20/20	167.7	77.6	0	20/20	235.7	117.9	0

Table 5: Results for rl5934

Pop size		5				10				15			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
EHBSA	WT/2	20/20	292.3	75.4	0	20/20	520.9	146.294	0	20/20	575.9	220.192	0
	WT/3	20/20	288.2	191.1	0	20/20	352.5	130.982	0	20/20	540.9	165.039	0
	WT/4	20/20	270.1	130.0	0	20/20	372.8	126.72	0	20/20	496.8	135.91	0
	WT/5	20/20	278.6	180.9	0	20/20	403.3	193.322	0	20/20	523.7	194.093	0
Pop size		15				30				60			
Operator		#OPT	MNE	STD	Error	#OPT	MNE	STD	Error	#OPT	MNE	STD	Error
OX		11/20	167.5	43.0779	0.0003	19/20	285.2	75.0882	0.0001	20/20	412.6	93.6421	0
PMX		15/20	159.5	34.0996	0.0000	17/20	276.9	62.3335	0.0000	20/20	516.4	99.2514	0
EER		17/20	220.6	47.4676	0.0000	20/20	408.7	84.4797	0	20/20	800.8	74.8484	0

cities problem

To summarize the results of hybridization with LK heuristic, we can say that EHBSA provided robust performance (over all population sizes) even for large TSP instances. The other recombination based methods provided performance that is inferior with respect to reliability and computational efficiency. #OPT found by EHBSA is always 100%; whereas the other operators needed larger population size for achieving this result. Thus we can say that the hybrid EHBSA works well for less population size, and is robust and needs less computational cost.

An experiment was also conducted to check if a particular template (e.g., the present best individual) would work better than a random template, but the results did not show much improvement.

4 Summary and Conclusions

A concept of edge histogram based sampling algorithm (EHBSA) for solving permutation problems was introduced by Tsutsui [12]. Its applicability to solve TSP [12] and flow shop-scheduling problems was also demonstrated [13].

In this paper EHBSA has been hybridized with some popular well-known local search heuristics (namely the 2-OPT and the Lin-Kerninghan heuristics) to take advantage of local search. Experimental evidences show that this hybrid technique decreases the computational requirements for finding the global optimal tour in TSP, providing a method capable of solving significantly larger problems of thousands of cities. The method is very robust in the sense of detecting global optimum for a wide variety of population sizes. The effectiveness of EHBSA, over other recombination-based techniques, is more clear when we use a weaker heuristic (like 2-OPT), whereas the effectiveness of EHBSA is less if we use a sophisticated heuristic like the Lin-Kerninghan heuristic. However, we can always observe some advantage of EHBSA over the classical recombination based techniques.

In future more in-depth empirical analysis would be performed to show that EHBSA combined with domain specific heuristic performs better in other permutation problems also.

Acknowledgments: This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant-in-Aid for Scientific Research number 13680469. A part of this work was done when Ashish Ghosh visited Hannan University, Osaka with a fellowship from Japan Society for Promotion of Science. Martin Pelikan was also partially supported by the Research Award and the Research Board at the University of Missouri.

References

[1] Bosman, P. A. N., & Thierens, D. (2001). Crossing the Road to Efficient ID*E*As for

Permutation Problems. *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2001*, 219-226.

[2] Dorigo M., Maniezzo, V. and Colorni, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. on Systems, Man, and Cybernetics-Part B*, Vol. 26, No. 1, 29-41.

[3] Goldberg, D.E. (1989). Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley.

[4] Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106-130.

[5] Larrañaga, P., & Lozano, J. A. (Eds.) (2002). Estimation of distribution algorithms: A new tool for evolutionary computation. Boston, MA: Kluwer Academic Publisher.

[6] Lin, S. & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498-516.

[7] Muehlenbein, H., & Paass, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature,* 178-187

[8] Oliver, I., Smith, D., and Holland, J. (1987). A study of permutation crossover operators on the travel salesman problem. *Proc. of the 2nd Int. Conf. on Genetic Algorithms*, 224-230.

[9] Pelikan (2002). Bayesian optimization algorithm: From single level to hierarchy. Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002023.

[10] Pelikan, M., Goldberg, D. E., & Lobo, F. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications, 21*(1), 5-20. Also IlliGAL Report No. 99018.

[11] Starkweather, T., McDaniel, S., Mathias, K, Whitley, D, and Whitley, C. (1991). A comparison of genetic sequence operators, *Proc. of the 4th Int. Conf. on Genetic Algorithms*, pp. 69-76Morgan Kaufmann.

[12] Tsutsui, S. (2002), Probabilistic Model-Building Genetic Algorithms in Permutation Representation Domain Using Edge Histogram, *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)*, pp. 224-233, Springer-Velag , Granada, 2002.9.

[13] Tsutsui, S., & Miki, M. (2002). Solving Flow Shop Scheduling Problems with Probabilistic Model-Building Genetic Algorithms using Edge Histograms. *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL02),* Singapore.

[14] Whitley, D., Starkweather, T., & Fuquay, D. (1989). Scheduling problems and traveling salesman problem: The genetic edge recombination operator. *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, Morgan Kaufmann.