

Probabilistic Model-Building Genetic Algorithms in Permutation Representation Domain Using Edge Histogram

Shigeyoshi Tsutsui

Department of Management and Information Science, Hannan University
5-4-33 Amamihigashi, Matsubara, Osaka 580-5802 Japan
tsutsui@hannan-u.ac.jp
<http://www.hannan-u.ac.jp/~tsutsui/>

Abstract. Recently, there has been a growing interest in developing evolutionary algorithms based on probabilistic modeling. In this scheme, the offspring population is generated according to the estimated probability density model of the parent instead of using recombination and mutation operators. In this paper, we have proposed probabilistic model-building genetic algorithms (PMBGAs) in permutation representation domain using *edge histogram based sampling algorithms (EHBSAs)*. Two types of sampling algorithms, without template (EHBSA/WO) and with template (EHBSA/WT), are presented. The results were tested in the TSP and showed EHBSA/WT worked fairly well with a small population size in the test problems used. It also worked better than well-known traditional two-parent recombination operators.

1 Introduction

Recently, there has been a growing interest in developing evolutionary algorithms based on probabilistic models [Pelikan 99b], [Larranaga 02]. In this scheme, the offspring population is generated according to the estimated probabilistic model of the parent population instead of using traditional recombination and mutation operators. The model is expected to reflect the problem structure, and as a result it is expected that this approach provides more effective mixing capability than recombination operators in traditional GAs. These algorithms are called *probabilistic model-building genetic algorithms (PMBGAs)* or *estimation of distribution algorithms (EDAs)*. In a PMBGA, better individuals are selected from an initially randomly generated population like in standard GAs. Then, the probability distribution of the selected set of individuals is estimated and new individuals are generated according to this estimate, forming candidate solutions for the next generation. The process is repeated until the termination conditions are satisfied.

Many studies on PMBGAs have been performed in discrete (mainly binary) domain and there are several attempts to apply PMBGAs in continuous domain. However, a few studies on PMBGAs in permutation representation domain are found. In this paper, we propose an approach of PMBGAs in permutation representation domain, and compare its performance with traditional recombination operators. In this approach, we develop an *edge histogram model* from the current population, where an edge is a link between two nodes in a string. We then sample nodes of a new string according to the edge histogram model. We will call this method the *edge histogram based sampling algorithm*

(EHBSA). We tested the algorithm in the Traveling Salesman Problem (TSP), a typical, well-known optimization problem in permutation representation domain. The results showed EHBSA worked fairly well on the test problems used. Section 2 of this paper gives a brief overview of PMBGAs. In Section 3, the two proposed EHBSAs are described. The empirical analysis is given in Section 4. Section 5 concludes the paper.

2 A Brief Overview of PMBGAs

According to [Pelikan 99b], these PMBGAs in binary string representation can be classified into three classes depending on the complexity of models they use; (1) no interactions, (2) pairwise interactions, and (3) multivariate interactions. In models with no interactions, variables are treated independently. Algorithms in this class work well on problems which have no interactions among variables. These algorithms include the PBIL [Baluja 94], cGA [Harik 98], and UMDA [Mhlenbein 96] algorithms. In pairwise interactions, some pairwise interactions among variables are considered. These algorithms include the MIMIC algorithm [De Bonet 97], the algorithm using dependency trees [Baluja 97]. In models with multivariate interactions, algorithms use models that can cover multivariate interactions. Although the algorithms require increased computational time, they work well on problems which have complex interactions among variables. These algorithms include ECGA [Harik 99] and BOA [Pelikan 99a, 00].

Studies to apply PMBGAs in continuous domains have also been made. These include continuous PBIL with Gaussian distribution [Sebag 98] and a real-coded variant of PBIL with iterative interval updating [Servet 97]. In [Gallagher 99], the PBIL is extended by using a finite adaptive Gaussian mixture model density estimator. The UMDA and MIMIC were introduced in continuous domain. All above algorithms do not cover any interactions among the variables. In EGNA [Larranaga,99], a Gaussian network learns to estimate a multivariate Gaussian distribution of the parent population. In [Bosman 99], two density estimation models, i.e., the normal distribution, and the histogram distribution are discussed. These models are intended to cover multivariate interaction among variables. In [Bosman 00a], it is reported that the normal distribution models have shown good performance. In [Bosman 00b], a normal mixture model combined with a clustering technique is introduced to deal with non-linear interactions. In [Tsutsui 01a, b], an evolutionary algorithm using marginal histogram models in continuous domain was proposed.

A study on PMBGAs in permutation domain is found in [Robles 02]. In it, PMBGAs are applied to solving TSP using two approaches. One is to use discrete PMBGAs and the other is to use continuous PMBGAs. In applying discrete PMBGAs, several different Bayesian network structures are compared and the *All Time Modification* (ATM) method was used to ensure that all the generated individuals were correct. In applying continuous PMBGAs, the correct tour is obtained by sorting the vectors of real numbers. PMBGAs are also applied to solve job shop scheduling problems and graph matching problems [Larranaga 02].

3 Edge Histogram Based Sampling Algorithm (EHBSA)

This section describes how the edge histogram based sampling algorithm (EHBSA)

can be used to (1) model promising solutions and (2) generate new solutions by simulating the learned model.

3.1 The Basic Description of the Algorithm

An *edge* is a link or connection between two nodes and has important information about the permutation string. Some crossover operators, such as Edge Recombination (ER) [Whitley 89] and enhanced ER (eER) [Starkweather, 91] which are used in traditional two-parent recombination, use the edge distribution only in the two parents string. The basic idea of the edge histogram based sampling algorithm (EHBSA) is to use the edge histogram of the whole population in generating new strings.

The algorithm starts by generating a random permutation string for each individual population of candidate solutions. Promising solutions are then selected using any popular selection scheme. An *edge histogram matrix (EHM)* for the selected solutions is constructed and new solutions are generated by sampling based on the edge histogram model. New solutions replace some of the old ones and the process is repeated until the termination criteria are met. This algorithm can be seen as a permutation version of the algorithm which uses marginal histogram models proposed in [Tsutsui 01a, b].

3.2 Developing Edge Histogram Matrix

Let string of k th individual in population $P(t)$ at generation t represent as $s_k^t = (\pi_k^t(0), \pi_k^t(1), \dots, \pi_k^t(L-1))$. ($\pi_k^t(0), \pi_k^t(1), \dots,$ and $\pi_k^t(L-1)$) are the permutation of $(0, 1, \dots, L-1)$, where L is the length of the permutation. Edge histogram matrix $EHM^t (e_{ij}^t)$ ($i, j = 0, 1, \dots, L-1$) of population $P(t)$ is symmetrical and consists of L^2 elements as follows:

$$e_{i,j}^t = \begin{cases} \sum_{k=1}^N (\delta_{i,j}(s_k^t) + \delta_{j,i}(s_k^t)) + \varepsilon & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad (1)$$

where N is the population size, $\delta_{ij}(s_k^t)$ is a delta function defined as

$$\delta_{i,j}(s_k^t) = \begin{cases} 1 & \text{if } \exists h [h \in \{0, 1, \dots, L-1\} \wedge \pi_k^t(h) = i \wedge \pi_k^t((h+1) \bmod L) = j] \\ 0 & \text{othersise} \end{cases} \quad (2)$$

and ε ($\varepsilon > 0$) is a bias to control *pressure* in sampling nodes just like those used for adjusting the selection pressure in the proportional selection in GAs. The average number of edges of element e_{ij}^t (ij) in EHM^t is $2LN/(L^2-L) = 2N/(L-1)$. So, ε is determined by a bias ratio B_{ratio} ($B_{\text{ratio}} > 0$) of this average number of edges as

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

A smaller value of B_{ratio} reflects the real distribution of edges in sampling of nodes and a bigger value of B_{ratio} will give a kind of perturbation in the sampling (see Section 3.3). An example of EHM^t is shown in Fig. 1.

Although we defined a symmetric EHM^t , i.e., $e_{i,j} = e_{j,i}$, which is applicable

$$\begin{array}{l} s_1^t = (0, 1, 2, 3, 4) \\ s_2^t = (1, 3, 4, 2, 0) \\ s_3^t = (3, 4, 2, 1, 0) \\ s_4^t = (4, 0, 3, 1, 2) \\ s_5^t = (2, 1, 3, 4, 0) \end{array} \quad \begin{array}{c} \left(\begin{array}{ccccc} 0 & 3.1 & 2.1 & 2.1 & 3.1 \\ 3.1 & 0 & 4.1 & 3.1 & 0.1 \\ 2.1 & 4.1 & 0 & 1.1 & 3.1 \\ 2.1 & 3.1 & 1.1 & 0 & 4.1 \\ 3.1 & 0.1 & 3.1 & 4.1 & 0 \end{array} \right) \end{array}$$

(a) $P(t)$
(b) EHM^t

Fig. 1. An example of symmetric edge histogram matrix for $N = 5$, $L = 5$, $B_{\text{ratio}} = 0.04$

to problems such as a symmetric TSP, but here we must note that we need to define an asymmetric EHM' for problems such as a asymmetric TSP or scheduling problems with permutation representation. An asymmetric EHM' can be easily defined by a equation similar to Eq. 1.

3.3 Sampling Methods

In this subsection, we describe how to sample a new string from the edge histogram matrix EHM' . We propose two types of sampling methods; one is an *edge histogram based sampling algorithm without template (EHBSA/WO)*, and the other an *edge histogram based sampling algorithm with template (EHBSA/WT)*.

3.3.1 Edge histogram based sampling algorithm without template (EHBSA/WO)

In EHBSA/WO, a new individual permutation $c[]$ is generated straightforwardly as follows:

1. Set the position counter $p \leftarrow 0$.
2. Obtain first node $c[0]$ randomly from $[0, L-1]$.
3. Construct a roulette wheel vector $rw[]$ from EHM' as $rw[j] = e^{c[p]j}$ ($j=0, 1, \dots, L-1$).
4. Set to 0 previously sampled nodes in rw ($rw[c[i]] = 0$ for $i=0, \dots, p$).
5. Sample next node $c[p+1] = x$ ($x \in \{y \mid rw[y] \neq 0\}$) 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.
8. Obtain a new individual string $c[]$.

Here, note that the EHBSA/WO is only applicable to problems where the absolute position of each node in a string has no meaning, such as in the TSP. This sampling method is similar in part to the sampling in Ant Colony Optimisation [Dorigo 96].

3.3.2 Edge histogram based sampling algorithm with template (EHBSA/WT)

EHM' described in Section 3.2 is in a marginal edge histogram. It has no explicit graphical structure. EHBSA/WT is intended to make up for this disadvantage by using a template in sampling a new string. In generating each new individual, a *template individual* is chosen from $P(t)$ (normally, randomly). The n ($n > 1$) cut points are applied to the template randomly. When n cut points are obtained for the template, the template should be divided into n segments. Then, we choose one segment randomly and sample nodes for the segment. Nodes in other $n-1$ segments remain unchanged. We denote this sampling method by EHBSA/WT/ n . Since average length of one segment is L/n , EHBSA/WT/ n generates new strings which are different at most L/n nodes on average from their templates. Fig. 2 shows an example

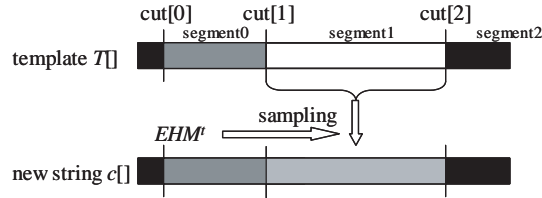


Fig. 2. An example of EHBSA/WT/3

of EHBSA/WT/3. In this example, nodes of new string from after cut[2] and before cut[1] are the same as the nodes of the template. New nodes are sampled from cut[1] up to, but not including, cut[2] based on the *EHM'*.

The sampling method for EHBSA/WT/*n* is basically the same as that of the EHBSA/WO as follows:

1. Choose a template $T[]$ from $P(t)$.
2. Obtain sorted cut point array $\text{cut}[0], \text{cut}[1], \dots, \text{cut}[n-1]$ randomly.
3. Choose a cut point $\text{cut}[l]$ by generating random number $l \in [0, n-1]$.
4. Copy nodes in $T[]$ to $c[]$ from after $\text{cut}[(l+1) \bmod n]$ and before $\text{cut}[l]$.
5. Set the position counter $p \leftarrow (\text{cut}[l] - 1 + L) \bmod L$.
6. Construct a roulette wheel vector $rw[]$ from *EHM'* as $rw[j] = e^{t_{c[p]j}}$ ($j=0, 1, \dots, L-1$).
7. Set to 0 copied and previously sampled nodes in $rw[]$ ($rw[c[i)] = 0$ for $i = \text{cut}[(l+1) \bmod n], \dots, p$).
8. Sample next node $c[(p+1) \bmod L] = x$ with probability $rw[x] / \sum_{j=0}^{L-1} rw[j]$ using roulette wheel $rw[]$.
9. Update the position counter $p \leftarrow (p+1) \bmod L$.
10. If $(p+1) \bmod L \neq \text{cut}[(l+1) \bmod n]$, go to step 6.
11. Obtain a new individual string $c[]$.

4. Empirical Study

4.1 Experimental Methodology

4.1.1 Evolutionary models

Here, we describe evolutionary models for EHBSA/WT, EHBSA/WO, and two-parent recombination operators, respectively. All these models are basically the same as *steady state models*.

(1) Evolutionary model for EHBSA/WT

Let the population size be N , and let it, at time t , be represented by $P(t)$. The population $P(t+1)$ is produced as follows (Fig. 3):

1. Edge distribution matrix *EHM'* described in Subsection 3.2 is developed from $P(t)$
2. A template individual $T[]$ is selected from $P(t)$ randomly.
3. EHBSA/WT described in Subsection 3.3.2 is performed using *EHM'* and $T[]$, and generate a new individual $c[]$.
3. The new $c[]$ individual is evaluated.
4. If $c[]$ is better than $T[]$, then $T[]$ is replaced with $c[]$, otherwise $T[]$ remains, forming $P(t+1)$.

(2) Evolutionary model for EHBSA/WO

Evolutionary model for EHBSA/WO is basically the same as the model for EHBSA/WT except EHBSA/WO does not use a template $T[]$. New string $c[]$ compares randomly selected individual

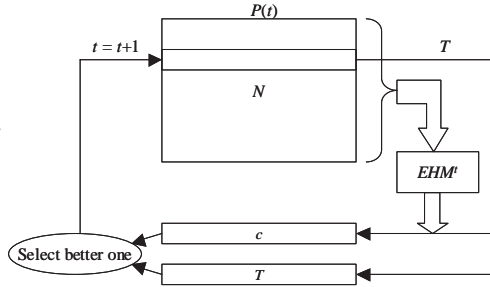


Fig. 3. Evolutionary model for EHBSA/WT

$i[]$ in $P(t)$, and if $c[]$ is better than $i[]$, $i[]$ is replaced with $c[]$.

(3) Evolutionary model for two-parent recombination operators

To compare the performance of proposed methods with the performance of traditional two-parent recombination operators, we designed an evolutionary model for two-parent recombination operators. For fair comparison, we design it as similar as possible to that of the EHBSA. We generate only one child from two parents. Using one child from two parents is already proposed for designing the GENITOR algorithm by Whitley et al. [Whitley 89]. In our generational model, two parents are selected from $P(t)$ randomly. No bias is used in this selection. Then we apply a recombination operator to produce one child. This child is compared with its parents. If the child is better than the worst parent, then the parent is replaced with the child.

4.1.2 Test suit and performance measures

We tested the algorithm in the Traveling Salesman Problem (TSP), a typical, well-known optimization problem in permutation representation domain. The following well-known data files have been used in this empirical study: 24 cities gr24, 48 cities gr48, and 76 cities pr76. The gr24 and gr48 are used in the study of TSP with EDA in [Robles 02]. We compared EHBSA with popular order based recombination operators, namely, the original order crossover OX [Oliver 87], the enhanced edge recombination operator eER [Starkweather 91], and the partially mapped crossover [Goldberg 89]. We also tried to compare EDBSA with results in [Robles 02] on gr24 and gr48.

Ten runs were performed. Each run continued until the optimum tour was found, the population was converged, or evaluations reached E_{\max} . Values of E_{\max} were 50000, 500000, and 1000000 for gr24, gr48, and pr76, respectively. Population sizes of 60, 120, 240 were used for EHBSA, and 60, 120, 240, 480, 960 for other operators, respectively. As to the bias ratio B_{ratio} in Eq. 3, B_{ratio} values of 0.03, 0.015, and 0.005 were used for gr24, gr48, and pr76, respectively.

We evaluated the algorithms by measuring their $\#OPT$ (number of runs in which the algorithm succeeded in finding the optimum tour), ANE (average number of evaluations to find the global optimum in those runs where it did find the optimum), and $Aver$ (average length of best solution in each run). Here, a lower value of ANE means a more effective search capability of an algorithm.

4.1.3 Blind search

In solving TSP using GAs, mutation operators play an important role. Several types of mutation operators are proposed. Also, it is well known that combining GAs with local optimization methods or heuristics greatly improve the performance of the algorithms. Many kinds of heuristics for TSP are proposed [Johnson 02]. For example, in [Ulder 90], Ulder et al. combined GAs with 2-opt heuristics and the algorithm showed greatly improved performance. In [Nagata 97], Nagata et al. proposed a high-power crossover operator for TSP which includes a kind of heuristics in the operator.

In this experiment, we use no mutation and no heuristic to see the pure effect of applying proposed algorithms. Thus, the algorithm is a *blind search*.

4.2 Empirical Analysis of Results

Results in gr24 are shown in Table 1. EHBSA/WO found the optimum tour 7, 9, and 6 times with $N = 60, 120,$ and $240,$ respectively. On the other hand, EHBSA/WT/ n found the optimum tour 10 times for all experiments. The ANEs of EHBSA/WT/2 and EHBSA/WT/3 were 9141, and 9523, respectively, showing good performance. Thus, we can see the performance of EHBSA/WT is much better than EHBSA/WO. In the other operators, eER showed good performance. The eER with $N = 240, 480,$ and 960 found the optimum tour 10 times and the ANE for $N = 240$ was 13394, which is a little larger than EHBSA/WT/ n with $N = 60.$ OX showed worse performance than eER although PMX showed the worst performance. Comparing the performance of EHBSA/WT with other operators, EHBSA/WT is slightly better than eER and is much better than OX and PMX. One big difference between EHBSA/WT and eER is that EHBSA/WT requires a smaller population size to work than eER. To compare EHBSA with results in [Robles 02] we see the results with discrete representation (here referred to as *discrete EDA*). In [Robles 02], it is shown that the discrete EDA without local optimization does not find the optimum tour.

Results in gr48 are shown in Table 2. EHBSA/WO could not find the optimum tour in gr48. On the other hand, EHBSA/WT/ n again found the optimum tour 10 times for all experiments except EHBSA/WT/4 and EHBSA/WT/2 with $N = 60.$ The ANEs for EHBSA/WT/3 and EHBSA/WT/5 were 85387 and 89799, respectively, showing good performance. Thus, we can see again the performance of EHBSA/WT is much better than EHBSA/WO. In the other operators, eER showed weaker performance than EHBSA/WT/ n in gr48, but better performance than OX. The best #OPT of eER is 5 with $N = 960$ and the ANE of this case is 166286, much larger than EHBSA/WT/ $n.$ PMX could not find the optimum tour. Comparing the performance of EHBSA/WT with discrete EDA in [Robles 02] is impossible because both termination conditions are different in gr48.

Results in pr76 are shown in Table 3. EHBSA/WO could not find the optimum tour in pr76. On the other hand, EHBSA/WT/ n found the optimum tour several times. With $N = 60,$ EHBSA/WT/2, 3, 4, and 5 found the optimum tour 4, 4, 9, and 10 times, respectively. With $N = 120,$ EHBSA/WT/2, 3, 4, and 5 found the optimum tour 9, 9, 9, and 10 times, respectively, showing the best performance. Thus, we can see the performance of EHBSA/WT is much better than the performance of EHBSA/WO in this experiment, too. In the other operators, eER found the optimum tour only 1 time with $N = 480$ and 3 times with $N = 960,$ showing worse performance than EHBSA/WT. OX and PMX could not find the optimum tour.

From the results described above, we can see that EHBSA/WT/ n worked fairly well in the test problems used. It also worked better than popular traditional two-parent recombination operators. In EHBSAs, the population size appears to be a crucial parameter as with traditional GAs. But one interesting feature of EHBSA/WT/ n is that it requires smaller population size than traditional two parent recombination operators. This may be an important property of EHBSA/WT/ $n.$ In our experiments, we used a blind search. When we combine EHBSA/WT/ n with some heuristics, it worked well with a smaller population size. As to the number of cut points, a smaller number of n work well with problems with smaller numbers of cities, and a larger number of n work

well with problems with larger numbers of cities; i.e., gr24: $n = 2$, gr48: $n = 3$, and pr76: $n = 5$.

Table 1. Results of gr24

Model	Population Size N														
	60			120			240			480			960		
	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver
EHBSA/WO	7	16328	1281	9	23637	1273	6	44853	1280	/			/		
EHBSA/WT/2	10	9141	1272	10	17978	1272	10	35604	1272						
EHBSA/WT/3	10	9523	1272	10	18251	1272	10	32956	1272						
EHBSA/WT/4	10	10677	1272	10	17652	1272	10	33606	1272						
EHBSA/WT/5	10	11170	1272	10	20489	1272	10	36078	1272						
OX	0	-	1345	1	22449	1303	4	34140	1296	1	48674	1301	0	-	1484
eER	1	4738	1299	7	6237	1276	10	13394	1272	10	23785	1272	10	42767	1272
PMX	0	-	1492	0	-	1414	2	23191	1341	1	49442	1316	0	-	1572
Other PMBGA*	#OPT=0, ANE is not available, best length = 1328 with MIMIC, best Aver = 1439 with EBNA														

Optimum: 1272

$E_{\max} = 50000$, $B_{\text{ratio}} = 0.03$

* Best data without heuristic using discrete EDA in [Robles 02]. Maximum evaluation is 50000

Table 2. Results of gr48

Model	Population Size N														
	60			120			240			480			960		
	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver
EHBSA/WO	0	-	5212	0	-	5316	0	-	5773	/			/		
EHBSA/WT/2	6	102691	5053	10	174125	5046	10	299391	5046						
EHBSA/WT/3	10	85387	5046	10	134597	5046	10	240391	5046						
EHBSA/WT/4	9	82701	5047	10	126444	5046	10	237260	5046						
EHBSA/WT/5	10	89799	5046	10	157041	5046	10	257486	5046						
OX	0	-	5527	0	-	5268	0	-	5200	1	162154	5099	2	287852	5082
eER	0	-	5653	0	-	5233	0	-	5098	2	95075	5072	5	166286	5058
PMX	0	-	8285	0	-	7374	0	-	6859	0	-	6116	0	-	5860
Other PMBGA*	#OPT=0, ANE is not available, best length = 6104 with MIMIC, best Aver = 6717 with MIMIC														

Optimum: 5046

$E_{\max} = 500000$, $B_{\text{ratio}} = 0.015$

* Best data without heuristic using discrete EDA in [Robles 02]. Maximum evaluation was set to 50000

Table 3. Results of pr76

Model	Population Size N														
	60			120			240			480			960		
	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver	#OPT	ANE	Aver
EHBSA/WO	0	-	119136	0	-	128208	0	-	142206	/			/		
EHBSA/WT/2	4	360128	108352	9	457147	108174	7	871319	108201						
EHBSA/WT/3	4	248091	108385	9	472719	108171	8	853801	108201						
EHBSA/WT/4	9	341482	108247	9	607544	108247	0	-	108496						
EHBSA/WT/5	10	494674	108159	10	797963	108159	0	-	108807						
OX	0	-	129603	0	-	121642	0	-	116591	0	-	113412	0	-	112259
eER	0	-	142003	0	-	122217	0	-	111839	1	-	109119	3	394887	108507
PMX	0	-	236827	0	-	213528	0	-	187601	0	-	164883	0	-	158515
Other PMBGA	not available														

Optimum: 108159

$E_{\max} = 1000000$, $B_{\text{ratio}} = 0.005$

5. Conclusions

In this paper, we have proposed probabilistic model-building genetic algorithms (PMBGAs) in permutation representation domain using the Traveling Salesman Problem (TSP), a typical, well-known optimization problem in permutation representation domain and compare its performance with traditional recombination operators. In this approach, we developed an edge histogram model from the current population. Two types of sampling algorithms, EHBSA/WO and EHBSA/WT, were presented. The results showed EHBSA/WT worked fairly well with a smaller size of population on the test problems used. It also worked better than well-known traditional two parent recombination operators.

There are many opportunities for further research related to the proposed algorithms. The effect of parameter values of B_{ratio} , number of cut point of the template n , and size of population N , on the performance of the algorithm must be further investigated. We experimented with EHBSAs using a blind search to test the pure mixing capability of the proposed algorithms. But we must test the algorithms with appropriate heuristics in problems with large numbers of cities. Analyzing the time complexity of the algorithm, and applying EHBSAs to other permutation problems, such as job shop scheduling problems, also remain for future work.

Acknowledgments

The authors gratefully acknowledge Prof. David E. Goldberg and Dr. Martin Pelikan for their valuable comments on PMBGAs during my stay at IlliGAL in 2001.

This research is partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant-in-Aid for Scientific Research number 13680469, and a grant to RCAST at Doshisha University from Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

- [Baluja 94] Baluja, S.: Population-based incremental learning: A method for interacting genetic search based function optimization and coemotive learning, *Tech. Rep. No. CMU-CS-94-163*, Carnegie Mellon University (1994).
- [Baluja 97] Baluja, S. and Davies: Using optimum dependency-trees for combinatorial optimization: learning the structure of the search space, *Tech. Rep. No. CMU-CS-97-107*, Carnegie Mellon University (1997)
- [Bosman 99] Bosman, P. and Thierens, D.: An algorithmic framework for density estimation based evolutionary algorithms, *Tech. Rep. No. UU-CS-1999-46*, Utrecht University (1999).
- [Bosman 00a] Bosman, P. and Thierens, D.: Continuous iterated density estimation evolutionary algorithms within the IDEA framework, *Proc. of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO-2000*, pp.197-200 (2000).
- [Bosman 00b] Bosman, P. and Thierens, D.: Mixed IDEAs, *Tech. Rep. No. UU-CS-2000-45*, Utrecht University (2000).
- [De Bonet 97] De Bonet, J. S., Isbell, C. L. and Viola, P.: MIMIC: Finding optima by estimating probability densities, In Mozer, M. C., Jordan, M. I., and Petsche, T. (Eds): *Advances in neural information processing systems*, Vol. 9, pp. 424-431. (1997).
- [Harik 98] Harik, G., Lobo, F. G., and Goldberg, D. E.: The compact genetic algorithm, *Proc. of*

- the Int. Conf. Evolutionary Computation 1998 (ICEC 98)*, pp. 523-528 (1998).
- [Harik 99] Harik, G: Linkage learning via probabilistic modeling in the ECGA, *Technical Report IlliGALReport 99010*, University of Illinois at Urbana-Champaign, Urbana, Illinois (1999).
- [Larranaga 99] Larranaga, P., Etxeberria, R., Lozano, J.A., and Pena, J.M.: Optimization by learning and simulation of Bayesian and gaussian networks, *University of the Basque Country Technical Report EHU-KZAAIK -4/99* (1999).
- [Dorigo 96] Dorigo M., Maniezzo, V. and Colorni, A.: The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Trans. on Systems, Man, and Cybernetics-Part B*, Vol. 26, No. 1, pp. 29-41 (1996).
- [Mhlenbein 96] Mhlenbein, H and Paa, G.: From recombination of genes to the estimation of distribution I. Binary parameters, *Proc. of the Parallel Problem Solving from Nature - PPSN IV*, pp. 178-187 (1996).
- [Pelikan 99a] Pelikan, M., Goldberg, D. E., and Cantu-Paz, E.: BOA: The Bayesian optimization algorithm, *Proc. of the Genetic and Evolutionary Computation Conference 1999 (GECCO-99)*, Morgan Kaufmann, San Francisco, CA (1999).
- [Pelikan 99b] Pelikan, M., Goldberg, D. E., and Lobo, F. G. : A survey of optimization by building and using probabilistic models, *Technical Report IlliGAL Report 99018*, University of Illinois at Urbana-Champaign (1999).
- [Pelikan 00] Pelikan, M., Goldberg, D. E., and Cantu-Paz, E.: Linkage problems, distribution estimate, and Bayesian network, *Evolutionary Computation*, Vol. 8, No. 3, pp. 311-340 (2000).
- [Sebag 98] Sebag, M. and Ducoulombier, A.: Extending population-based incremental learning to continuous search spaces, *Proc. of the Parallel Problem Solving from Nature - PPSN V*, pp. 418-427 (1998).
- [Servet 97] Servet, I. L., Trave-Massuyes, L., and Stern, D.: Telephone network traffic overloading diagnosis and evolutionary computation techniques, *Proc. of the Third European Conference on Artificial Evolution (AE 97)*, pp. 137-144 (1997).
- [Larranaga 00] Larranaga, P., Etxeberria, R., Lozano, J. A., and Pena, J. M.: Optimization in continuous domains by learning and simulation of Gaussian networks, *Proc. of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 201-204 (2000).
- [Robles 02] Robles, V., Miguel, P. D., and Larranaga, P.: Solving the traveling salesman problem with EDAs, *Estimation of Distribution Algorithms*, Larranaga, P. and Lozano, J. A. (eds), Kluwer Academic Publishers, Chapter 10, pp. 211-229 (2002).
- [Larranaga 02] Larranaga, P. and Lozano, J. A. (eds): *Estimation of distribution algorithms*, Kluwer Academic Publishers (2002).
- [Tsutsui 01] Tsutsui, S., Pelikan, M., and Goldberg, D. E.: Evolutionary Algorithm using Marginal Histogram Models in Continuous Domain, *Proc. of the 2001 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 230-233 (2001).
- [Nagata 97] Nagata, Y. and Kobayashi, S.: Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem, *Proc. of the 7th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, pp. 450-457 (1997).
- [Johnson 02] Johnson, D. S, and McGeoch, L. A.: Experimental analysis of heuristics for the STSP, *The Traveling Salesman Problem and its Variations*, Gutin and Punnen (eds), Kluwer Academic Publishers, Chapter 1 (to appear).
- [Oliver 87] Oliver, I, Smith, D., and Holland, J.: A study of permutation crossover operators on the travel salesman problem, *Proc. of the 2nd Int. Conf. on Genetic Algorithms*, pp. 224-230 (1987).
- [Starkweather, 91] Starkweather, T., McDaniel, S., Mathias, K, Whitley, D, and Whitley, C.: A comparison of genetic sequence operators, *Proc. of the 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, pp. 69-76 (1991).
- [Whitley 89] Whitley, D., Starkweather, T., and Fuquay, D.: Scheduling problems and traveling salesman problem: The genetic edge recombination operator, *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, Morgan Kaufmann (1989).
- [Ulder 90] Ulder, N., Pesch, E., van Laarhoven, P., Bandelt, and Aarts, E.: Improving TSP exchange heuristics by population genetics, *Proc. of the Parallel Problem Solving from Nature - PPSN* (1990).