

Entropy-based Genetic Algorithm for Solving TSP

*Yasuhiro TSUJIMURA and Mitsuo GEN

Dept. of Industrial and Information Systems Engg., Ashikaga Institute of Technology

Abstract — The Traveling Salesman Problem (TSP) is used as a paradigm for a wide class of problems having complexity due to the combinatorial explosion. TSP has become a target for the Genetic Algorithm (GA) community, because it is probably the central problem in combinatorial optimization and many new ideas in combinatorial optimization have been tested on the TSP. However, by using GA for solving TSPs, we obtain a local optimal solution rather than a best approximate solution frequently.

The goal of this paper is to solve the above mentioned problem about local optimal solutions by introducing a measure of diversity of populations using the concept of information entropy. Thus, we can obtain a best approximate solution of the TSP by using this Entropy-based GA.

Keywords: Traveling Salesman Problem, Genetic Algorithms, Diversity of Population, Information Entropy

1 Introduction

The *Traveling Salesman Problem* is one of the most widely studied combinatorial optimization problems. Its statement is deceptively simple: a salesman seeks the shortest tour through n cities, with each city visited exactly once before returning back to his hometown. No disconnected subtour is allowed.

The TSP belongs to the class of *NP-hard*. A *Genetic Algorithm*(GA)[1][2] is a powerful tool for solving combinatorial optimizing problems, and some applications of GAs to TSPs have been reported[3]-[7]. Applying GA to combinatorial optimization problems, however, we frequently encounter a trap of falling into a local optimal solution. And the quality of such a local solution may

be not so good as we expect. This phenomena occurs when the diversity of a population at the later generation decrease. Decrement of diversity population causes deterioration of effects of the genetic operation.

In this paper, we introduce the concept of the information entropy as a measure of diversity of each population into the process of GA to solve the above mentioned problem of falling into a local optimal solution. And we call this GA introduced the measure of diversity *Entropy-based GA*(EBGA). We can obtain a best approximate solution of the TSP problem of FFM using this EBGA. Furthermore, we verify efficiency of the EBGA by numerical experiments.

2 Implementation of GA for TSP

Basic implementation of GA for solving a TSP is shown in this section. This GA implementation basically employs the *simple GA*[3]-[4]. The main issues of the GA, chromosome representation, evaluation function, and genetic operators, are given in the following subsections.

2.1 Representation Scheme

Path representation scheme[2] are employed as a suitable one for representing a schedule of the original problem. A chromosome t_k ($k = 1, 2, \dots, pop_size$; pop_size is the population size) is represented as:

$$t_k = (c_1 \ c_2 \ \dots \ c_n)$$

where

c_i : the i th city to be visited, $i = 1, 2, \dots, n$

Example: Considering 6 cities TSP, a route: 1-2-6-4-3-5 can be represented using the path representation as (1 2 6 4 3 5)

*Corresponding author
268 Ohmae-cho, Ashikaga 326, JAPAN
Phone: +81(284)62-0605, Fax: +81(284)64-1071
E-mail: tujimr@ashitech.ac.jp

2.2 Evaluation Function

The fitness gives main pressure for convergence of evolution process. It is calculated by the evaluation function. The evaluation function $eval(t_k)$ ($k = 1, 2, \dots, pop_size$) is defined as:

$$eval(t_k) = \frac{1}{\sum_{i=1}^n d(c_i, c_{i+1}) + d(c_n, c_1)} \quad (1)$$

where

$d(c_i, c_{i+1})$: travelling distance from city c_i to city c_{i+1}

2.3 Genetic Operators

There are three main genetic operations, crossover operation, mutation operation, and selection. The path (permutation-based) representation is often used because it is much suitable for representing the solutions of many kinds of combinatorial optimization problem.

So a variety of crossover operation have been developed for the path representation, such as *partially mapped crossover* (PMX)[1][2], *order crossover* (OX)[1][4], *cycle crossover* (CX)[1], and so on. CX was employed in this implementation of GA.

The *swap mutation operation*[1][2] is introduced into our GA. Regeneration process is controlled under the *roulette wheel selection* mechanism[1][2].

3 Entropy-based Genetic Algorithm

In this section, to avoid falling into a local optimal solution in the evolution process we propose *Entropy-based GA* (EBGA) which promises to provide good convergentability of evolution process. The EBGA can keep high diversity of a chromosome population.

3.1 Measure of Diversity of a Chromosome Population

For GA applications, falling into a local optimal solution in the evolution process is one of the most important problems should be solved. Increment of number of similar chromosomes in a population spoils the ability of mutation operation to keep high diversity of the population, and then falling into the trap of local optimal solution is often raised[8].

In the EBGA, to escape from the trap of local optimal solution, we measure the diversity of the chromosome population at every generation, and then improve low diversity populations. Measurement of the total diversity of a population is accomplished by obtaining the locus diversity for every loca of all chromosomes in the population (see Figure 1).

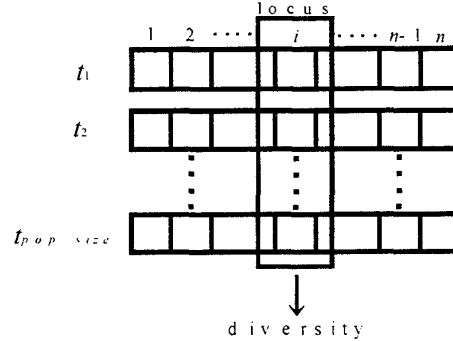


Figure 1: The concept of the evaluation of a population diversity

The locus diversity H_i of the i th locus ($i = 1, 2, \dots, n$) is defined as

$$H_i = - \sum_{c \in C} pr_{ic} \ln pr_{ic} \quad (2)$$

where

$$pr_{ic} = \frac{na_{ic}}{pop_size}$$

C : the set of cities should be visited

na_{ic} : the number of appearance of city c at locus i

The locus diversity H_i is derived from the concept of the *information entropy* [9], and it can be used as a measure of the diversity of the i th locus of a chromosome population. H_i approaches to the maximum value ($\ln pop_size$) when each city in C appears uniformly in the population, conversely it approaches to the minimum value 0 when the number of appearances of one of the cities in C is much more than the numbers of appearances of the others.

The locus diversity H_i is evaluated by comparing with the threshold value. Here, we use $\ln 2$ as the threshold value. Taking $\ln 2$ for the threshold value means "At the locus, just two kinds of cities appear pop_size times". If $H_i \leq \ln 2$, the locus i has low diversity. We compare every loca with the threshold value, and if the number of loca, which has the locus diversity below the threshold value, is greater than or equal $\frac{n}{2}$, the diversity of the chromosome population is too low. Then, for such a low diversity population, we rise the diversity of the popu-

lation using the procedure shown in the following subsection. a is the control parameter which takes an integer value from the closed interval $[2, 5]$. The value of a is decided by the relation between the population size and the number of cities in C , and the larger the value of a is, the higher the probability of doing the improvement is.

3.2 Improvement of Diversity of a Population

A chromosome population which has been considered the low diversity one by using the procedure mentioned in Subsection 3.1, should be improved to be an enough high diversity population. The improvement of the population diversity is performed according to the following procedure:

[Procedure: Mod-Pop]

Step 1 : Select m chromosomes from the population.

m is a random integer number in $[\frac{pop_size}{a}, pop_size - 1]$.

Step 2 : In each of the chromosomes selected, exchange genes among the loca which have lower locus diversities than the threshold value.

After doing the procedure, we evaluate the population diversity again using the method described in the previous subsection. Repeat the process mentioned above until getting enough high population diversity.

Example: In case of $n = 5$ and $pop_size = 5$, the following population is given.

	locus i				
	1	2	3	4	5
$t_1 =$	(4	1	5	2	3)
$t_2 =$	(5	4	1	2	3)
$t_3 =$	(4	2	1	5	3)
$t_4 =$	(5	3	1	4	2)
$t_5 =$	(4	3	5	1	2)

Using eq.(2), the entropy of each locus in this population is calculated as follows:

Comparing H_i with the threshold value $\ln 2 \approx 0.693$ and setting the control parameter $a = 3$, the loca 1, 3, 5 are recognized as the low diversity loca as follows:

$i \backslash k$	1	2	3	4	5
1	$\frac{0}{5}$	$\frac{1}{5}$	$\frac{3}{5}$	$\frac{1}{5}$	$\frac{0}{5}$
2	$\frac{0}{5}$	$\frac{1}{5}$	$\frac{0}{5}$	$\frac{2}{5}$	$\frac{2}{5}$
3	$\frac{0}{5}$	$\frac{2}{5}$	$\frac{0}{5}$	$\frac{0}{5}$	$\frac{3}{5}$
4	$\frac{3}{5}$	$\frac{1}{5}$	$\frac{0}{5}$	$\frac{1}{5}$	$\frac{0}{5}$
5	$\frac{2}{5}$	$\frac{0}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{0}{5}$
	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
	H_1	H_2	H_3	H_4	H_5
	\parallel	\parallel	\parallel	\parallel	\parallel
	0.673	1.332	0.673	1.332	0.673

Figure 2: The concept of the evaluation of a population diversity

	locus i				
	1	2	3	4	5
$t_1 =$	(4	1	5	2	3)
$t_2 =$	(5	4	1	2	3)
$t_3 =$	(4	2	1	5	3)
$t_4 =$	(5	3	1	4	2)
$t_5 =$	(4	3	5	1	2)

We apply the 'Procedure: Mod-Pop' to this population

Step 1: Chromosomes t_1 , t_2 and t_5 have been selected randomly.

	locus i				
	1	2	3	4	5
$t_1 =$	(4	1	5	2	3)
$t_3 =$	(4	2	1	5	3)
$t_5 =$	(4	3	5	1	2)

Step 2: In each of the selected chromosomes, exchange genes between loca 1, 3, 5.

$t_1 =$	(4	1	5	2	3)	\longrightarrow	$t'_1 =$	(3	1	4	2	5)
$t_3 =$	(4	2	1	5	3)	\longrightarrow	$t'_3 =$	(4	2	3	5	1)
$t_5 =$	(4	3	5	1	2)	\longrightarrow	$t'_5 =$	(2	3	5	1	4)

As the result, we can obtain the following improved population:

	locus i				
	1	2	3	4	5
$t'_1 =$	(3	1	4	2	5)
$t'_2 =$	(5	4	1	2	3)
$t'_3 =$	(4	2	3	5	1)
$t'_4 =$	(5	3	1	4	2)
$t'_5 =$	(2	3	5	1	4)

3.3 Overall Procedure of EBGa

Overall procedure of EBGa is shown below.

[Procedure: EBGa]

Step 0: Setting the parameters

Set the parameters: number of cities n , population size pop_size , crossover probability pr_c , mutation probability pr_m , and maximum generation $maxgen$.
Let generation $gen = 0$, $maxeval = 0$

Step 1: Initialization

Generate pop_size chromosomes randomly.

Step 2: Evaluation

Step 2.1: Calculate the fitness value of each chromosome using eq.(1).

Step 2.2: If

$maxeval < \max\{eval(t_k)\}$

then

$s^* = \text{argmax}\{eval(t_k)\}$

and

$maxeval = \max\{eval(t_k)\}$

where 'argmax' is the argument maximum operator which takes the t_k having the largest fitness value.

Step 3: Crossover

Perform the crossover CX on chromosomes selected with probability pr_c .

Step 4: Mutation

Perform the swap mutation on chromosomes selected with probability pr_m .

Step 5: Selection

Select pop_size chromosomes from the parents and offspring for the next generation by the roulette wheel selection method.

Step 6: Improvement of the population diversity

Step 6.1: Calculate the locus diversity of each locus of the population by eq.(2).

Step 6.2: Let $count = 0$ and $I = \{\phi\}$.

Compare $H_i (i = 1, 2, \dots, n)$ with the threshold value $\ln 2$.

If $H_i \leq \ln 2$, then add the index i to the set I and let $count = count + 1$.

Repeat this step from $i = 1$ to n .

Step 6.3: If $count \geq \frac{n}{a}$, then go to

Step 6.4, else go to **Step 7**.

Step 6.4: Generate a random integer number $rand$ from

$[\frac{pop_size}{a}, pop_size - 1]$.

Select $rand$ chromosomes from the population randomly.

In the selected chromosomes, exchange genes randomly between the loca included in the set I (Procedure: Mod-Pop).

Go to **Step 6.1**.

Step 7: Stop testing

If $gen = maxgen$, then output s^* and stop, else let $gen = gen + 1$ and return to **Step 2**.

4 A Numerical Experiment and Discussion

We tested the EBGa by comparing with traditional GA which skips Step 6 in **Procedure: EBGa** using a 20 cities TSP of which the distance data is given in Appendix.

All of the tests were accomplished on NEC PC-9821Ap with i486 CPU. With the following parameter setting:

population size $pop_size = 20$
maximum generation $maxgen = 20000$
crossover probability $pr_c = 0.8$
mutation probability $pr_m = 0.2$
control parameter $a = 3$

the convergence processes, obtained by averaging among 20 runs, of EBGa and GA are shown in Figure 3 in which the similar process of the traditional GA is also shown for comparison.

Seeing Figure 3, we recognized that the EBGa can provide better solution the traditional GA, and also it has much faster convergentability than the GA has in average.

5 Conclusions

The trap of falling into a local solution, *i.e.*, prematurity is one of the most important and critical problems should be solved on applications of GA to the TSPs.

In this paper, we proposed the Entropy-based GA (EBGa) which improves the population diversity by introducing the concept of the information entropy to escape from the local optimal trap. And practical use of the EBGa demonstrated by a numerical experiment and comparison with the traditional GA. As the results, it was made clear that the EBGa can obtain a good solution with shorter execution time than the GA spends to get the same quality solution.

For future research works, we will apply the EBGA to combinatorial optimization problems lying in industrial engineering, such as shop scheduling problems, design of delivery systems, optimal computer network design, and so on.

References

- [1] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd eds., Springer-Verlag, 1994.
- [2] Gen, M. and R. Cheng: *Genetic Algorithms and Engineering Design*, John Wiley & Sons, 1997.
- [3] Grefenstette, J., R. Gopsl, B. J. Rosmaita and D. Van Gucht: Genetic Algorithms for the Traveling Salesman Problem, *Proc. of the 1st ICGA*, pp.160-168, 1985.
- [4] Gen, M., Y. Tsujimura, K. Takamura and R. Cheng: Numerical Experiments on Methods for Solving the Traveling Salesman Problem by GA, *Proc. of 1993 Fall Meeting of JIMA*, pp.33-34, 1993 (in Japanese).
- [5] Whitley, D., T. Starkweather and D. Fuquay: Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator, *Proc. of 3rd ICGA*, pp.133-140, 1989.
- [6] Yamamura, M., T. Ono, and S. Kobayashi: Character-Preserving Genetic Algorithms for Traveling Salesman Problem, *Journal of Japanese Society for AI*, Vol.7, No.4, pp.1049-1059, 1992 (in Japanese).
- [7] Tsutsui, S. and Y. Fujimoto: The f-GA: Forking Genetic Algorithm with Blocking and Shrinking Modes, *Journal of Japanese Society for AI*, Vol.9, No.5, pp.741-747, 1994 (in Japanese).
- [8] Mori, K., M. Tsukiyama and T. Fukuda: Immune Algorithm with Searching Diversity and Its Application to Resource Allocation Problem, *Trans. of IIE of Japan*, Vol.113-C, No.10, pp.872-878, 1993 (in Japanese).
- [9] Kapur, J. N. and H. K. Kesavan: *Entropy Optimization Principles with Application*, Academic Press, 1992.

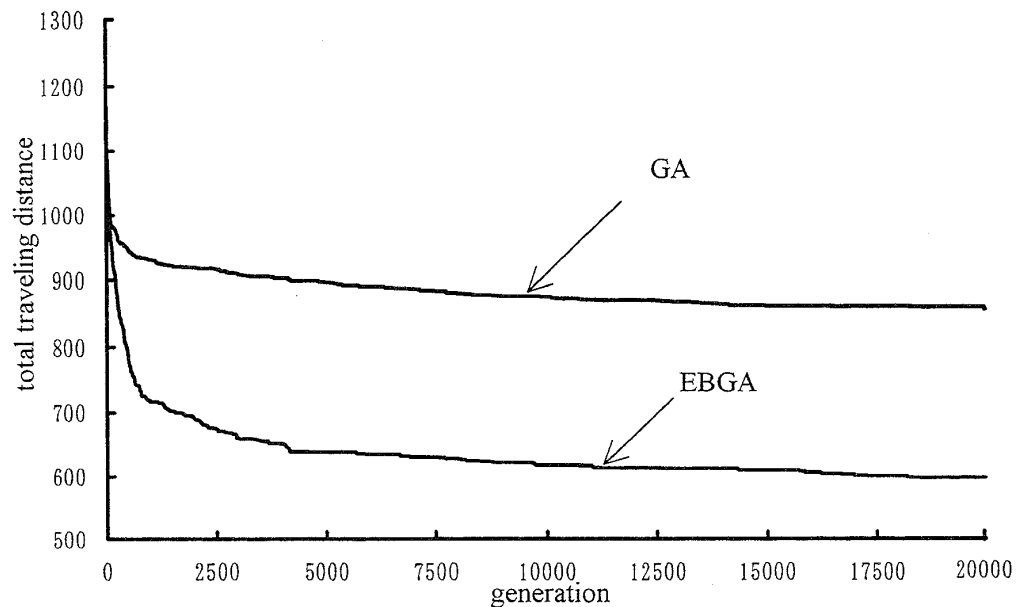


Figure 3: Convergent process

Appendix: The distance data of the 20 cities TSP

1:Arlon																			
40	2:Bastgne																		
63	65	3:Bouillon																	
99	59	81	4:Durbuy																
144	104	169	74	5:Eupen															
128	88	114	33	68	6:Huy														
57	17	89	51	87	84	7:Houfflize													
69	29	72	26	112	59	25	8:Laroche												
126	86	129	48	45	33	69	74	9:Liege											
26	70	89	125	141	158	87	99	158	10:Luxemburg										
115	75	128	60	29	91	58	51	47	112	11:Malmedy									
83	43	65	16	92	49	45	20	64	113	63	12:Mons								
36	28	37	72	132	105	45	50	120	62	103	56	13:Namur							
61	34	47	47	120	80	42	25	95	87	100	31	25	14:St. Hubert						
94	54	119	75	50	108	37	62	68	91	21	82	82	79	15:St. Uith					
134	79	134	41	33	61	62	62	35	131	19	57	122	87	40	16:Spa				
126	66	151	43	44	76	49	51	52	123	15	48	94	76	32	24	17:Stavelot			
152	94	140	59	15	63	77	80	30	149	37	75	130	105	58	18	42	18:Verviers		
28	68	52	111	172	144	84	89	159	51	145	95	39	64	121	162	53	315	19:Virton	
144	104	137	65	47	51	87	92	18	170	55	82	132	112	86	46	70	29	151	20:Vise