An Efficient Genetic Algorithm with Fuzzy c-Means Clustering for Traveling Salesman Problem

Jong-Won Yoon and Sung-Bae Cho
Dept. of Computer Science
Yonsei University
Seoul, Korea
jwyoon@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract—Genetic algorithms (GA) are one of effective approaches to solve the traveling salesman problem (TSP). When applying GA to the TSP, it is necessary to use a large number of individuals in order to increase the chance of finding optimal solutions. However, this incurs high evaluation costs which make it difficult to obtain fitness values of all the individuals. To overcome this limitation we propose an efficient genetic algorithm based on fuzzy clustering which reduces evaluation costs with minimizing loss of performance. It works by evaluating only one representative individual for each cluster of a given population, and estimating the fitness values of the others from the representatives indirectly. A fuzzy c-means algorithm is used for grouping the individuals and the fitness of each individual is estimated according to membership values. The experiments were conducted with randomly generated cities, and the performance of the method was evaluated by comparing to other GAs. The results showed the usefulness of the proposed method on the TSP.

Keywords—genetic algorithm, fitness evaluation, fuzzy clustering, fuzzy c-means algorithm

I. INTRODUCTION

The traveling salesman problem (TSP) [1] is a well-known non-deterministic polynomial-time (NP)-hard problem in combinatorial optimization. Given a list of \( N \) cities \( V = \{v_1, v_2, \ldots, v_N\} \) and distances between them \( W = \{w_{11}, w_{12}, \ldots, w_{1N}, \ldots, w_{NN}\} \) where \( w_{ij} \) indicates the distance from \( v_i \) to \( v_j \), the goal of the TSP is to find a shortest tour path that visits each city only once. Fig. 1 shows examples of the TSP.

![Figure 1. Examples of the traveling salesman problem (a: Cities to be visited, b: A possible tour path but not optimal, c: The shortest possible tour path)](image)

![Figure 1. Examples of the traveling salesman problem (a: Cities to be visited, b: A possible tour path but not optimal, c: The shortest possible tour path)](image)

The TSP has been applied to various real-world problems, such as logistics, data clustering, computing genome sequences [1]. Since it is one of NP-hard optimization problems, there are only approximation algorithms to solve. A genetic algorithm (GA) is one of well-known effective approaches to solve the TSP [2] since it is known that it has higher chance to get a global optimum than other searching or optimization methods [3].

When using GA to solve the TSP, it is important to maintain the size of the population as large as possible to search the global optimal solution. This causes some problems when applying GA to the problem because it takes too much cost to evaluate the fitness of a large number of individuals in the population. This problem is related to not only the size of the population but also the cost taken to evaluate an individual. In order to surmount the limitations, many researchers suggested evaluating individuals partially, for example, the modified interactive GA that the users evaluate only the part of individuals and the rest of individuals are evaluated by the computer is proposed [4]. It is still an important issue how to estimate the fitness values as precise as possible while reducing the cost of the evaluation.

In this paper, we applied an efficient GA that partially evaluates and estimates fitness values by using the fuzzy clustering technique to the TSP to overcome the limitations of the conventional GA. The proposed method aims to keep the similar performance to the standard GA. In addition, we conducted experiments with randomly generated TSP data to verify that the method can apply to the TSP successfully.

II. EFFICIENT GENETIC ALGORITHMS

In general, it is difficult to define the fitness evaluation function and almost impossible to evaluate by human as the size of the population grows in the case of the interactive GA. For the inverse problem of the engineering similarly the fitness evaluation takes much time and cost [5]. In order to reduce the costs of the fitness evaluation in the GA, many researchers have proposed efficient GAs that evaluate only the part of the population and estimate the fitness of the rest individuals. These methods had its origin in the characteristics of the GA that the optimization is achieved as a generation is proceeding based on individuals.

One of them works proposed was a GA that evaluates only a few individuals directly and estimates the remainder of the individuals by examining their similarities to the selected individuals [6]. In addition, a hybrid GA based on clustering has been developed. This GA can considerably reduce the number of evaluations by evaluating only one representative of each cluster’s center after clustering all the individuals in the...
given population [7, 8]. Graening et al. suggested that the best individual of each cluster be re-evaluated using the real fitness function [7]. Jin and Sendhoff evaluated the only the cluster centers with the real fitness functions and other individuals with a neural network ensemble [8]. However, such hard clustering techniques that distribute the fitness value of each of the individuals linearly with only one representative individual to which each individual belongs do not provide accurate fitness values unless the clustering forms an ideal cluster partition.

For the efficient GAs, it is important to prove that the algorithms can be applied to the practical problems successfully. Even though some efficient GAs has been proposed, they have been verified with only some simple benchmark functions, including our previous works [9, 10], without any applications to complex and practical problems.

In this paper, we proposed to apply an efficient GA with the fuzzy clustering technique that can estimate fitness values more precisely with fuzzy membership-based soft cluster boundaries to solve the TSP with less costs.

III. FUZZY CLUSTERING-BASED EFFICIENT GA FOR TSP

In this paper, we propose an efficient GA for the TSP using the fuzzy clustering technique which shows performance to the conventional GA even though it has fewer costs for fitness evaluations.

It is constructed with three parts: In the first part, the individuals are grouped with their similarities and membership values are obtained. Secondly, the centroid of each cluster is evaluated by the original fitness function and the fitness values of other individuals are estimated with the membership values in the next part. Fig. 2 shows the flow of the proposed method. Finally, the general GA operations are performed. This produces a new generation and the whole process is repeated until the maximum number of generations is reached.

A. Designing GA for TSP

Prior to apply the proposed efficient GA to the TSP, it is necessary to design GA for the problem. First of all, we applied random key encoding (RKE) scheme [11] to encode chromosomes. RKE is a strategy available for problems involving permutation evolution [12], and it can represent the TSP also [11]. Moreover, in RKE, the genes are represented by real numbers and these are suitable to the proposed method since it uses the clustering technique (See next section). In RKE, a real number is assigned for each gene (and it is mapped to each city). In decoding, the order of visiting cities is generated as the ascending order of their corresponding numbers. Fig. 3 shows an example of RKE for the TSP. As shown on the figure, for example, the third city which has 0.16 in the chromosome is visited firstly after the decoding because 0.16 is the smallest number in the chromosome. On the contrary, the second city which has 0.75 is visited lastly since it is the highest value among all the cities.

![Figure 3. An example of RKE for the TSP](image)

Since standard crossover techniques can be used for RKE, we used one-point crossover. The fitness value of ith individual, \( f_i \), was defined as below:

\[
 f_i = \frac{1}{w_{c_1(i)c_2(i)} + w_{c_2(i)c_3(i)} + \ldots + w_{c_K(i)c_1(i)} }
\]  

(1)

where \( c_k(i) \) represents the kth visited city in the solution of ith individual. The denominator of Eq (1) means the length of the tour path. The shorter the path is, the greater the corresponding fitness value is.

B. Fuzzy Clustering

In order to separate individuals into several groups, a fuzzy clustering algorithm is used for grouping the population instead of a hard clustering algorithm. A fuzzy clustering approach is less likely to get stuck in the local minimum than a hard clustering approach since it makes soft decisions in iteration through the use of membership values.

The most widely-used fuzzy clustering algorithm is the fuzzy c-means algorithm, proposed by Bezdeck [13]. It generates a fuzzy partition that provides each piece of data with a degree of membership to a given cluster. The values of the degrees of membership lie between 0 and 1. Values close to 0 indicate the absence of association to the corresponding cluster, while values close to 1 indicate strong association to the cluster. Fig. 2 shows the procedure of the fuzzy c-means algorithm.

C. Fitness Estimation

Since we evaluate only the centroids with the original fitness function, fitness values of remaining individuals should
be estimated. Fitness estimation is rather important and it is necessary to use an appropriate and efficient fitness estimation method because performance depends on accurate fitness estimation of the individuals. Fig. 4 shows an instance of the fitness estimation process with the centers of the clusters constructed by the clustering algorithm.

1) Determine the number of clusters \( c \) and the fuzziness parameter \( m \)
2) Initialize the membership matrix \( u_{ij} \) satisfying the condition:
   \[
   \sum_{i=1}^{c} u_{ij} = 1, \quad 1 \leq j \leq n
   \]
3) Compute centroids \( v_i \) (\( i = 1, 2, \ldots, c \)):
   \[
   v_i = \frac{\sum_{j=1}^{n} u_{ij} x_j}{\sum_{j=1}^{n} u_{ij} ^m}
   \]
4) Compute membership values matrix \( U \):
   \[
   \mu_{ij} = \frac{\left( \frac{1}{d^m(x_i, v_j)} \right)^{\frac{1}{m-1}}}{\sum_{k=1}^{c} \left( \frac{1}{d^m(x_i, v_k)} \right)^{\frac{1}{m-1}}}
   \]
5) Compute the objective function \( J_m \):
   \[
   J_m(X, U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} (u_{ij})^m d^2(x_i, v_i)
   \]
6) Repeat 3) through 5) until stabilized as:
   \[
   ||f_m^{(i)} - f_m^{(i-1)}|| \leq \varepsilon
   \]

Figure 4. The fuzzy c-means clustering algorithm

As shown in Fig. 5, suppose that \( S = \{s_1, s_2, \ldots, s_n \} \) is a set of individuals in the population, \( C = \{C_1, C_2, \ldots, C_c \} \) is a set of clusters, and the fitness values of the cluster centers are \( F = \{f_1, f_2, \ldots, f_c \} \). The fitness values of an individual can be estimated based on the similarity between all the centers of the clusters and the individual. \( m_{ij} \) indicates the degree of similarity between the \( i \)th individual and the \( k \)th cluster center.

For optimal fitness estimation, a fuzzy integral [14] is used to calculate the similarity between the individuals. The fuzzy integrals are the integrals of a real function with respect to a fuzzy measure, compared to the Lebesgue integral, which is defined with respect to an ordinary measure. Hence fuzzy integration constitutes a vast family of aggregation operators, including many widely-used ones suitable for this kind of aggregation. We have adopted this fuzzy integral to calculate the individual’s similarity measure \( m_{ij} \) over the centers of all the clusters.

Let \( h_k: S \rightarrow [0,1] \) be the degree of belongingness of an individual to the \( k \)th cluster, where 1 indicates absolute certainty that the individual is in the \( k \)th cluster and 0 implies absolute certainty that the individual is not in the \( k \)th cluster. The similarity measure \( m_{ij} \) between the \( i \)th individual and the \( k \)th cluster center is as follows:

\[
m_{ij} = h_k(s_i)
\]

and the distributed fitness value of \( s_i \) is as follows:

\[
s_i = \int h_k(s_i) \times f_j \sum_{s \neq i} h_k(s_i) \times f_j
\]

\[
= \sum_{s \neq i} m_{ij} \times f_i
\]

Since the number of clusters is discrete, the fuzzy integral of \( k \) could be substituted by just using the sum of the values. Also \( h_k(s_i) \) could be substituted by \( m_{ij} \) using the Eq (3).

**D. Computational Complexity**

The main advantage of the proposed method is that it reduces a computational complexity. In this section, we analyzed the reduction of the complexity with the big-\( O \) notation. Since the conventional GA and the proposed method share same GA procedures except fitness evaluations, only the complexity of the fitness evaluation step was treated.

The complexity of the fitness evaluation for a single generation of the conventional GA can be defined as follows:

\[
GA = nO(f) = O(n^f)
\]

where \( n \) is the number of individuals and \( O(f) \) is the complexity of the fitness evaluation function. On the other hand, the complexity of the same part of the proposed method is defined as below:

\[
GA_{FCM} = O(ndc^2i) + O(cf) + O(n)
\]

\[
= O(max(ndc^2i,cf,n))
\]

where \( d \) is the length of the chromosome – in the case of the TSP, it is related to the number of cities, \( c \) is the number of clusters, and \( i \) is required number of iterations for the clustering. Eq (5) consists of three terms. The first term is the computational complexity of the fuzzy c-means clustering algorithm [15]. The second term shows the complexity for evaluating total \( c \) centroids of clusters. The last third term represents the fitness estimation of \( n \) individuals, and this can be completed in \( O(n) \) without any extra computations since the fitness values of centroids and membership values of individuals were already obtained. By the summing-rule of the big-\( O \) notation, the complexity is determined as the maximum value among three terms. However, because \( d, c, \) and \( i \) are
greater than 0, \( n \) cannot be greater than \( ndc^2i \). Therefore, only
the first and the second term are chosen as the computational
complexity of the proposed method.

![Figure 6. Total running steps of GA and FCM depending on the length of chromosomes \((n = 100, c = 10, i = 50, O(f) = O(d^4))\)]

Since the number of clusters \( c \) is smaller than the number of
individuals \( n \), The proposed method is faster than the
conventional GA if \( O(ndc^2i) \subseteq cf \). Even though
\( O(ndc^2i) \supset O(cf) \), the proposed method can reduce
complexity if \( O(ndc^2i) \subset O(nf) \) which means that the
computational complexity of fitness evaluations of the
conventional method is greater than the complexity for the
fuzzy clustering.

Fig. 6 shows the total running steps depending on the length of
chromosomes when the complexity fitness function is
dependent on it. We fixed \( n, c, i \) as 100, 10, 50 respectively,
and changed \( d \) from 10 to 350. The complexity of fitness
function was set to \( O(d^4) \). The conventional GA is presented as
GA, and FCM is the proposed method. If \( d \) is small, GA
requires less total running steps than FCM. However, as \( d \)
grows, FCM runs faster than GA since the fitness function
which is dependent on \( d \) requires more running steps.

Fig. 7 shows the steps depending on the complexity of
fitness function. We fixed \( n, c, i, d \) as 100, 10, 50, 10 respectively.
When the fitness function is rather simple, GA
performs faster than FCM. On the other hand, FCM requires
less steps than GA if the fitness function is complex enough
that \( O(ndc^2i) \subset O(nf) \) (in this case, \( O(f) = O(d^4) \)).

IV. EXPERIMENTAL RESULTS

We conducted experiments to prove the usefulness of the
proposed method on the TSP. In order to show an outstanding
performance of the method, we compared several GA methods
including existing efficient GAs with partial evaluation.

A. Experimental Settings

For the experiments, total 30 cities were used and distances
between them were set randomly. The crossover rate and the
mutation rate were 0.75 and 0.0005, respectively.

Several GA methods – including some methods which
evaluate individuals partially – were used to compare
performances. Simple GAs with population sizes of 100 (Pop100)
and 10 (Pop10), partially evaluated GAs with hard
clustering techniques including the single linkage (S-L), the
hard c-means (HCM), and the k-means(KM), and the proposed
method (FCM) were compared. The number of cluster \( c \) was
set to 10. All experiments were conducted 20 runs and the
average results were used. The general parameters of the GA
are shown in Table 1.

<table>
<thead>
<tr>
<th>Table I. Parameters in the experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop100</td>
</tr>
<tr>
<td>Population size</td>
</tr>
<tr>
<td># of evaluations</td>
</tr>
<tr>
<td># of clusters</td>
</tr>
<tr>
<td>Length of chromosome</td>
</tr>
<tr>
<td>Crossover rate</td>
</tr>
<tr>
<td>Mutation rate</td>
</tr>
<tr>
<td># of generations</td>
</tr>
<tr>
<td>Fuzziness parameter</td>
</tr>
<tr>
<td>Terminal condition</td>
</tr>
</tbody>
</table>

B. Experimental Results

Table 2 shows the distance of shortest tour path obtained
from Pop100, Pop10, the hard clustering algorithms; S-L,
HCM, and KM, and the proposed method (FCM). FCM
performed showed even better performance than the alternative
methods except Pop100. Only HCM performed better than in
Pop10 by estimating the fitness of individuals even though they
evaluated the same number of individuals, however, worse than
in the FCM. The result implies that the proposed method
estimates the fitness more accurately.

<table>
<thead>
<tr>
<th>Table II. The distance of the shortest tour path from Pop100, Pop10, three hard clustering algorithms and the FCM (average of 20 runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop100</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Error</td>
</tr>
</tbody>
</table>

Fig. 8 shows the evolution process for the TSP with several
methods. As a result, Pop10 evolved slowly and never reached
the optimal solution. Among the hard clustering algorithms,
only KM produced better solutions than. HCM showed better
performance than Pop10 for a moment, however, only in the
beginning of the evolution. The FCM showed the best
performance among the other alternative methods except
Pop100 and this implied that the proposed method is more
accurate with the fuzzy membership based estimation process
described in section III.B. Even though the FCM evolved more
slowly than Pop100, it showed the most similar result with reduced complexity cost.

Figure 8. Evolution processes for the TSP problem

Table 3 shows the results of the paired t-test between the conventional method (Pop100) and the alternative methods. The t-value represents how different the performances from Pop100 and each comparison methods are. Although the statistics accepted the hypothesis that there are differences between the results from Pop100 and all the alternative methods since absolute t-value of each method is greater than the two-sided critical value of t, the FCM showed the lowest difference between t-value and the critical value which implied that the FCM showed the most similar result to Pop100 than any alternative methods. It turns out to be very efficient in terms of time and costs to get similar results without evaluating all the 100 individuals.

Table 4 shows the comparison result of the evaluation time required for the two methods – Pop100 and FCM. The time was reduced to almost one eighth via the proposed method. This result implied the usefulness of the proposed method that can provide accurate fitness estimation with less number of evaluations in the population.

V. CONCLUDING REMARKS

We proposed an efficient genetic algorithm (GA) for the traveling salesman problem (TSP). The method requires less fitness evaluation due to the process of fuzzy clustering. This process divides the whole population into several clusters, and evaluates one individual for each cluster. The fitness values of the others are estimated from the fitness values of the representative individuals indirectly by their membership functions. Results from experiments confirm that the algorithm produces the most dominant performance than other methods and it reduces computational complexity.

However, there still exist gaps in performance between the conventional GA and the proposed method. In order to narrow these gaps, enhanced estimation methods should be investigated in the future. The method should be also applied to extended benchmark data sets of TSP and more real-world problems which actually take high cost to evaluate fitness.

REFERENCES