Dynamic Diversity Control by Injecting Artificial Chromosomes for Solving TSP Problems

Pei-Chann Chang, Wei-Hsiu Huang, Julie Yu-Chih Liu, Cycer Chen, Ching-Jung Ting

Abstract—**The applications of genetic algorithms (GAs) in solving combinatorial problems are frequently faced with a problem of early convergence and the evolutionary processes are often trapped in a local but not global optimum. This premature convergence occurs when the population of a genetic algorithm reaches a suboptimal state that the genetic operators can no longer produce offspring with a better performance than their parents. In the literature, plenty of work has been investigated to introduce new methods and operators in order to overcome this essential problem of genetic algorithms. As these methods and the belonging operators are rather problem specific in general. In this research, we take a different approach by observing the progress of the evolutionary process and when the diversity of the population dropping below a threshold level then artificial chromosomes with high diversity will be introduced to increase the average diversity level thus to ensure the process can jump out the local optimum. The proposed method is implemented independently of the problem characteristics and can be applied to improve the global convergence behavior of genetic algorithms. The experimental results using TSP instances show that the proposed approach is very effective in preventing the premature convergence when compared with the earlier approaches.**

I. Introduction

The fundamental principles of GAs were first presented by Holland. Since that time, GAs have been successfully applied to a wide range of problems including multimodal function optimization, machine learning, combinatorial optimization problems and the evolution of complex structures such as neural networks. An overview of GAs and their implementation in various fields is given by Goldberg [6] or Michalewicz [1].

The advantage of applying GAs to hard combinatorial optimization problems lies in the ability to search the solution space in a broader way than heuristic methods based upon neighborhood search. Nevertheless, GAs are also frequently faced with a problem of stagnating in a local but not global optimum. This drawback, called premature convergence of GAs, occurs when the population of a GA reaches such a suboptimal state that the genetic operators can no longer produce offspring with a better performance than their parents.

As Ursem [2] pointed out in 1999, the diversity measure is traditionally used to analyze the evolutionary algorithms rather than guide them. However, a new application by adaptive controlling; that measuring and using different properties of the swarm/population while running, adds significant potential to the algorithm. In this research, we have therefore adopted the idea from Ursem [2] with the decreasing and increasing diversity operators to control the population diversity. Therefore, a proper balance between exploration and exploitation search can be maintained by controlling the diversity level of the population. The control mechanism can be built into the GA, and the idea is to control the diversity of the population by injecting Diversified Artificial Chromosomes (DAC) into the system until the diversity measure reaches a certain level than stop. The modified model uses a diversity measure and artificial chromosomes generation mechanism to control the evolutionary processes alternating between exploring and exploiting behavior. By monitoring the diversity of the population, once the diversity is above a certain threshold d_{low}, the regular evolutionary process takes in control. When the diversity declines below d_{low} the model simply inject DAC to the system until the threshold d_{high} is met again.

II. Traveling Salesman Problem

Through the years the Traveling Salesman Problem has occupied the thoughts of numerous researchers. There are several reasons for this. Firstly, the TSP is very easy to describe, yet very difficult to solve. No polynomial time algorithm is known with which it can be solved. This lack of any polynomial time algorithm is a characteristic of the class of NP complete problems, of which the TSP is a classic example. Second, the TSP is broadly applicable to a variety of routing and scheduling problems. Thirdly, since a lot of information is already known about the TSP, it has become a kind of "test" problem; new combinatorial optimization methods are often applied to the TSP so that an idea can be formed of their usefulness. Finally, a great number of problems actually treated with heuristic techniques in Artificial Intelligence are related with the search of the best permutation of n elements. We refer to the research of P. Larrañaga [15] for introduction TPS.

Pei-Chann Chang is with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C. (Corresponding Author's E-mail: iepchang@saturn.yzu.edu.tw).

Wei-Hsiu Huang is with the Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C.

Julie Yu-Chih Liu is with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C.

Cycer Chen is with the Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C.

Ching-Jung Ting is with the Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C.

978-1-4244-1823-7/08/$25.00©2008 IEEE

The TSP objective is to find the shortest route for a traveling salesman who, starting from his home city, has to visit every city on a given list precisely once and then return to his home city. The main difficulty of this problem is the immense number of possible tours, this is a problem in discrete or combinatorial optimization. It is a prominent illustration of a class of problems in computational complexity theory which are classified as NP-hard. Therefore, we apply TSP to do the feasibility study which will be applied the future research in solving problem of production scheduling.

III. A Dynamic Diversity Control Genetic Algorithm

Population diversity is a key issue in the performance of evolutionary algorithms. A proper control of the population diversity is needed during the evolutionary processes to balance the exploration and exploitation search, thus, a global optimum can be more likely reached through the mechanism of DDCGA. A common hypothesis is that high diversity is important for the process of algorithm to avoid premature convergence and escaping from local optimal solution. In the literatures survey, various diversity measures have been used to analyze algorithms, but so far few algorithms applied these measure approaches to guide the search directions.

The basic idea of this research is to measure the diversity level during the evolutionary processes and once the diversity of the population drops down to the threshold level, then the system will be introduced a certain level of artificial chromosomes with high diversity into the mating pool. Therefore, the diversity level of the population will be increased up to a certain degree; the evolutionary processes will consistently reduce the diversity level again. However, the control mechanism will ensure the artificial chromosomes be re-introduced once the diversity level drops below the threshold level. The proposed method for injecting diversity of chromosome is implemented independently from the problem characteristics to improve the global convergence behavior of genetic algorithms.

There are totally two phases in the process of DDCGA; one is SGA, when population diversity is higher than threshold, the fitness will be introduced to the traditional GA mechanism, another phase is DAC which will enhance the population diversity through injecting diversified artificial chromosomes into the mating pool.

The architecture of a Dynamic Diversity Control Genetic Algorithm is shown in Figure 1. As the process observed in the evolutionary process, when the fitness reaches to a local optimum, the chromosomes within the last couples of population will be very homogenous. Thus, the genetic operators cannot further generate better chromosomes to jump out the local optimality. It also indicates that the diversity of chromosome at this moment is very low or reaches the threshold value of d_{low}. DDCGA will trigger an artificial chromosome control mechanism to generate a set of chromosomes with high diversity and inject these chromosomes into the population. Although, the fitness of these artificial chromosomes may not as good as those within the original population, however, the diversity of these AC is pretty high. Hopefully, the exploration of the evolutionary process can restart again. In the ultimate goal, a better chromosome with good fitness can be generated in the long run.

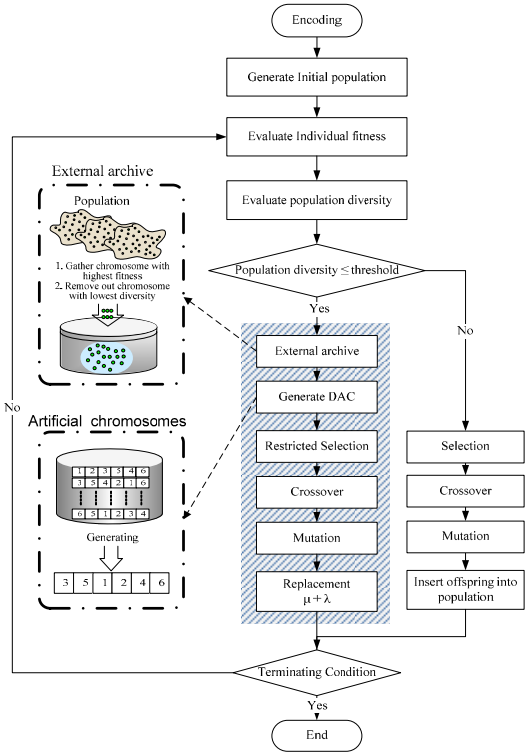

Fig. 1. The Architecture of DDCGA

In order to generate an artificial chromosome with good diversity, we need to breed a set of seed chromosomes before hand and these seed chromosomes are retrieved from the earlier evolutionary process. We establish a breeding pool by keeping the top 100 chromosomes with a higher diversity while the fitness is still acceptable. The artificial chromosomes will be generated from the set of breeding pool when the diversity of the population reaches a low threshold value. These artificial chromosomes are injected into the evolutionary process and the genetic operators will be able to generate new generations hopefully will lead the process to jump out the local traps. Hence, the new generation will simultaneously keep the exploration and exploitation features which ensure to obtain the best fitness in the long run.

Consequently, DAC is integrated into the procedure of Genetic Algorithm and it attends to improve the diversity performance of Genetic Algorithm. The primary procedure is to collect gene information first from the external archive and to use the gene information to generate artificial chromosomes. Before collecting the gene information, AC collects the chromosomes whose fitness is better by

comparing the fitness value of each chromosome with average fitness value of current population. Thus, the average fitness is calculated.

There are two segments in DDCGA, one is SGA which is applied when population diversity larger than threshold, another part is DDCGA which is to enhance the population diversity through inject artificial chromosomes into the mating pool. The pseudo code of the DDCGA algorithm is listed as follows:

Population: The population used in the Genetic Algorithm
Generations: The number of generations
Termination Condition:

1. Initialize *Population*
2. while termination condition is not satisfied do
3. Evaluate fitnesses and diversities of population
4. if population diversity > threshold then
5. Selection operator
6. Crossover operator
7. Mutation operator
8. else
9. Create artificial chromosomes
10. Select parents and artificial chromosomes applied by Selection operator
11. Crossover operator
12. Mutation operator
13. $\mu + \lambda$ Replacement
14. end if
15. end while

IV. An Artificial Chromosome Generation Mechanism

As we discussed in the above section, AC mechanism will influence the convergence of fitness, therefore the facilitated AC mechanism is extraordinarily for GA to extend searching space. AC here in the research is integrated into the procedure of genetic algorithm and it attends to improve the diversity of genetic algorithm. The primary procedure is to gather gene information by archive and to use the gene information to generate artificial chromosomes. Before gathering the gene information into archive, it gathers the chromosomes whose fitness is current best for whole iterations. When an archive size equal to the population size, then it gathers a chromosome with current best into archive and removes a chromosome with lowest diversity in archive.

After we gather gene information into dominance matrix, we assign genes to the positions for each artificial chromosome. The assignment sequence for every position is assigned randomly, which is able to inject diversify for artificial chromosomes. Thereupon, we select one gene for assigning to each position by roulette wheel selection method, based on the probability of each gene appearing in each position, the appearing frequency will be considered to the built dominance matrix. After we assign one gene to a position, the gene and position in the dominance matrix will be removed. Then, the process repeats to select the next gene until all genes are assigned.

We firstly suppose a discrete problem with 5 assigned genes, and the population size is set to ten, which means an archive will gather ten chromosomes from adaptation process. Moreover, we accumulate the gene information from these ten chromosomes into dominance matrix. For the position 1, there are two appearing frequency for gene 1, two for gene 2, two for gene 3, one for gene 4, and three for gene 5. Therefore, the dominance matrix contains the gene information of better chromosomes in position which is shown in Figure 2.

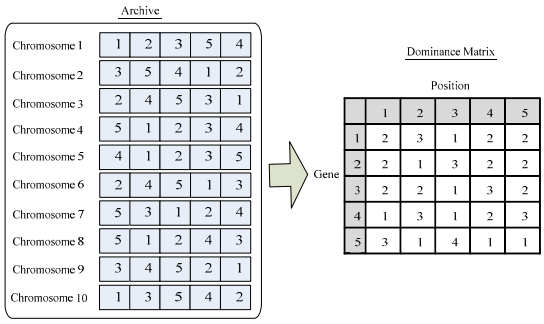

Fig. 2. The transferring process of Dominance Matrix

The appearing frequency of each gene is recorded to the corresponding position for gathering the information for computation of statistics. After all the appearing frequencies of genes are recorded, we transferred to these statistical data into each artificial chromosome. A probability matrix is further derived from the dominance matrix by the following formula:

$$P_{ij}(t) = \frac{\sum_{k=1}^{N} X_{ij}^{k}}{N} \tag{1}$$

Where $P_{ij}(t)$ is the probability of job *i* to be assigned in position *j*.

$$P(t) := \begin{bmatrix} P_{11}(t) & \cdots & P_{1n}(t) \\ \vdots & \ddots & \\ P_{n1}(t) & \cdots & P_{nn}(t) \end{bmatrix} \tag{2}$$

Base on the above Dominance Matrix, we will assign the first gene at position 3 in the beginning, which is shown in Figure 3. The frequency of each gene at position 3 is SEQ(1, 3, 1, 1, 4) from gene 1 to job 5. As the result of the number of total frequency is 10, the corresponding probability for gene 1 is 1/10; gene 2 is 3/10, and so on. Hence, we accumulate the probability from job 1 to 5 and utilize roulette wheel selection to apply this accumulated probability which is shown in Figure 3.

Gene	Pos. 3	Prob.	Accum
1	1	1/10	1/10
2	3	3/10	4/10
3	1	1/10	5/10
4	1	1/10	6/10
5	4	4/10	10/10

Fig. 3. The Probability Matrix from Dominance Matrix transformation

In order to enhance the exploration ability of the proposed algorithm, the $\mu + \lambda$ replacement strategy is applied. μ is the parent chromosomes and λ is the artificial chromosomes generated from archive. We combine artificial chromosomes into the mating pool, the $\mu + \lambda$ replacement strategy which combines previous parent population and artificial chromosomes. Afterward, we select μ , better chromosomes from the mating pool. Consequently, higher diversity solutions are preserved to the next generation.

V. DIVERSIFIED ARCHIVE DESIGN

For the reason to control AC, we have to control the generator mechanism. In this research, the main control mechanism is the Archive; therefore, the efficiency of Archive design will influence the convergence speed of fitness, and the fitness quality. Generally, for the concept of dynamic injecting diversity of chromosome, population diversity is a key issue in the performance of evolutionary algorithms. Basically, a proper control of the population diversity is needed during the evolutionary processes to balance the exploration and exploitation searching. Thus, a global optimum can be more likely reached. A common hypothesis is that high diversity is important for avoiding premature convergence and to escape local optimal solution. Various diversity measures have been used to analyze algorithms, but so far few algorithms have used a measure approach to guide the search.

The basic idea of this research is to measure the diversity level during the evolutionary processes and once the diversity of the population drop down to a threshold level then the system will introduce a certain level of artificial chromosomes with high diversity into the mating pool. Therefore, the diversity level of the population will be increased up to a certain degree. Of course, the evolutionary processes will reduce the diversity level again. However, the control mechanism will ensure the artificial chromosomes will be re-introduced once the diversity level drops below the threshold level. The proposed method is implemented independently of the problem characteristics and can be applied to improve the global convergence behavior of genetic algorithms. In this chapter, we will go to detail about how to achieve a dynamic diversity control in genetic algorithm and how to apply this in combinatorial problems.

There are several measures and methods have been used to promote population diversity. These methods typically use a nonstandard selection, mating, or replacement strategy or dynamic adopting the probability of genetic operator to increase or control diversity. In this paper, we propose a novel method, dynamic diversity control in genetic algorithm (DDCGA), of promoting diversity by artificial chromosomes.

A. Individual Chromosome Diversity

Diversity has been an important concept in ecological theory and application. Under various names, it appears in several biological, physical, social, management and engineering sciences. In the biology, diversity refers to differences between individuals and a population, which implies a structural and behavioral. But in the genetic algorithm, diversity refers to structural differences only.

We have discussed about the major problem in traditional GAs. The applied approach to measure diversity between chromosomes and injection timing is extremely important to overcome this kind of premature problems. Therefore, we arrange the researches proposed to the following formulas. We will firstly introduce the measure approach, then, discuss the design of experiment to test the relative parameters, which will be shown in the coming section.

The diversity of the population during the evolutionary process is set according to the following diversity-measure:

$$diversity(P) = \frac{1}{|L| \cdot |P|} \sum_{i=1}^{|P|} \sqrt{\sum_{j=1}^{N} \left(s_{ij} - \overline{s_j} \right)^2} \tag{3}$$

Where $|L|$ is the length of the diagonal1 in the search space, P is the population, $|P|$ is the population size, N is the dimensionality of the problem, s_{ij} is the j'th value of the i'th individual, and $\overline{s_j}$ is the j'th value of the average point s .

The DGEA applies diversity-decreasing operators (selection and recombination) as long as the diversity is above a certain threshold d_{low}. When the diversity drops below d_{low}, the DGEA switches to diversity-increasing operators (mutation) until a diversity of d_{high} is reached. Hence, phases with exploration and phases with exploitation will occur. Theoretically, the DGEA should be able to escape local optima because the operators will force higher diversity regardless of fitness.

If $d_{low} = d_{high,}$ the algorithm will maintain a diversity close to the given threshold value, which is particularly useful for dynamic and multi-objective optimization tasks.

An important issue is to apply a mutation operator that rather quickly increases the distance-to-average-point measure. Otherwise, the algorithm will stay in “explore"-mode for a long time. A straightforward idea for a measure-increasing mutation operator is to use the average point of the population to calculate the direction of each individual's mutation. The individual is then mutated with the Gaussian mutation operator, but now with a mean directed away from the average point. The purpose of this mutation operator is to force the individuals away from the population center. Preliminary results indicated that scaling the

normalized direction vector by 0.001 turned out to give the best results.

B. Population Diversity Measures

Diversity measures are highly problems depended, in different kinds of combination problems, we need to find some methodologies to measure the individual diversity between the best fitness and individual chromosome, denotes *X*. Others chromosomes denote set ***Y*** and *L* is the length of a chromosome.

In different kinds of combinatorial problems, diversity measures are tightly problems depended. There are two types to measure individual diversity in combinatorial problems. One is measure the difference between two genotypes. Bossert described a structural difference measure based on mathematics foundation. Let one of individual chromosomes as the best fitness chromosome denotes *X*, and the others chromosomes denote set *Y* and *L* is the length of a chromosome. As researches surveyed in the literatures, there are five diversity measure approaches proposed which are listed as follow:

1. By hamming distance:

The hamming distance is used between two strings of equal length which is the number of positions for which the corresponding symbols are different. We use the representation as a permutation way in GA to solving the combination problem, such as TSP or scheduling problem. To measure the individual diversity, we compare each gene with the best fitness chromosome and others chromosomes. *I* is an indicate function which is defined as the total number of positions where $x_i \neq y_j$.

Definition: There are two kinds of diversity measure denote $D(X,Y)$ by hamming distance below:

$$D(X,Y)=\frac{I}{L}, I=\sum_{j=0}^{L} I_j, I_j=\begin{Bmatrix} x_j = y_j, 0 \\ x_j \neq y_j, 1 \end{Bmatrix} \tag{4}$$

$$D(X,Y)=1-\frac{\sum_{i=0}^{L}(x_i - y_i)}{M} \tag{5}$$

Where $M=\begin{Bmatrix} L^2 - 1/2, \text{ if } L \text{ is odd.} \\ L^2/2, \text{ if L is even.} \end{Bmatrix}$

2. By Euclidean distance:

Euclidean distance is used to a real encoding, the concept is the same with Hamming distance in permutation encoding.

$$D(X,Y)=\sqrt{\sum_{i=1}^{N}(x_i + y_i)^2} \tag{6}$$

3. By connection matrix:

Considering a TSP problem, each tour represents as a permutation way in GA. Therefore, the diversity measures by hamming distance cannot reflex a true touring situation in TSP, and the connection matrix is considered the sequence in a tour. Although each tour represents as a permutation way differently in GA, there are still some chances those touring sequence are the same in connection matrix.

Definition:

$$A=\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0(n-1)} \\ a_{10} & a_{11} & & \\ \vdots & & \ddots & \vdots \\ a_{(n-1)0} & & \cdots & a_{(n-1)(n-1)} \end{bmatrix} \tag{7}$$

Where *A* is connection matrix of a tour, n=number of cities. Let a similarity function *S(X,Y)* measure the similarity.

$$S(X,Y)=\sum_{i,j}(x_{ij} \mid x_{ij}=y_{ij}=1)/n \tag{8}$$

Where n is the numbers of cities. So that the diversity measure could be defined as follow:

$$D(X,Y)=1-S(X,Y) \tag{9}$$

4. By information entropy:

In information theory, the Shannon entropy or information entropy is a measure of the uncertainty associated with a random variable. It quantifies the information contained in a message, usually in bits or bits/symbol. The locus diversity H_i of the i_{th} locus (i = 1,…,n) is defined as follow:

$$H_i \equiv -\sum_{c \in C} pr_{ic} \ln pr_{ic}, \text{ where } pr_{ic}=\frac{na_{ic}}{pop\_size} \tag{10}$$

Where na_{ic}: the number of appearance of city *c* at locus *I*, *C*: the number of cities should be visited.

We need to translate the individual diversity to a single colony index to measure the population diversity is low or high. There are two kinds of method to measure it.

a. Arithmetic average

$$PD=\frac{\sum D(X,Y)}{N} \tag{11}$$

b. Linear scale measure

$$PD=\frac{\overline{d}-d_{\min}}{d_{\max}-d_{\min}} \tag{12}$$

Where $\overline{d}$ is the average of individual diversity, d_{max} is the maximum individual diversity and d_{min} is the minimum individual diversity. Artificial chromosomes are developed according to the observation and a dominance matrix which will record this gene distribution information in archive. The dominance matrix is transformed into a probability matrix to decide the next assignment of a gene to a position.

C. Archive Design

For seeking best parameter combinations, we firstly list the influence coefficient to be the factors to be analyzed in Table I. In this research, for the reason to validate the adaptability of DDCGA, we do not consider the interaction for seeking the maximum improvement; moreover, the factors we experiment would not influence the efficiency of other factors. Therefore, we adopt one factorial design to analyze.

TABLE I
DESIGN OF EXPERIMENT FOR ARCHIVE

Factor	A	B	C	D	E
Archive Population Policy	Fitness	Diversity			
Archive Cage Size	50	100	150	200	
Dynamic Fitness Adjustment	None	With			
Diversity Threshold (10^{-2})	0.01	0.03	0.05	0.07	0.09
Diversity Threshold (10^{-1})	0.1	0.2	0.3	0.4	0.5

From the factors in table above, we will experiment for several factors, the first three factors will be considered in this section, and the last two will be compared in the section of experimental result. It was used in GAs for keeping a beneficial chromosomes when GAs adaptation, but in this research, it was used by gathering diversify chromosomes.

Therefore, suppose a process in which random genetic changes generated by mutation and crossover explores the genotype space. Occasionally, these explorations stumble onto a significant innovation. These innovations can bestow such an advantage that the population of the new genotype explodes, generating an episode of mass extinction as it drives other genotypes out of memory. The extinction episode is noted as a sharp drop in the diversity. Thus, it drops appear to correspond to the chance discovery of significant innovations.

However, continued mutation and crossover operator new variants of the successful new form. This process generally restores the community to the equilibrium diversity about as rapidly as the diversity was lost in the extinct allel. In this research, we would like to gather diversity for each generation. Therefore, we design two main mechanism of archive in DDCGA. Both of these mechanisms gather a representative of chromosome into archive, and then removing an ineffective chromosome when archive is loaded. The most significant difference is, one is designed for sieving out the chromosome with better fitness in Archive, and another Archive design is to sieve out the chromosomes with higher diversity. The concept of these two philosophies is indicated in Figure 4.

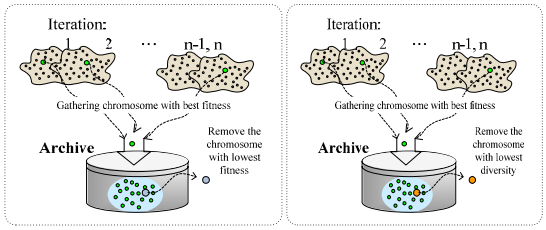

Fig. 4. Illustration of Archive infrastructure

Firstly, a chromosome with best fitness will be gathered, which is the protagonist with best fitness in an evolution. When a current best chromosome is changed, another one current best chromosome is gathered by archive. But we cannot infinitely gather chromosomes; moreover, we need to determine an exchange mechanism for removing an ineffective chromosome out of archive.

We therefore design two criterions for removing an ineffective chromosome. Fitness-based Archive is to remove the lowest fitness chromosome in archive, which means that the chromosome with better fitness will be kept in the Archive for generating better artificial chromosome. The policy of Diversity-based Archive is to remove the lowest diversity in archive, which the main idea is to keep the chromosome with higher diversity, the higher diversity represents the searching space is significantly different from the space is been searched now. Therefore, we could regard the chromosome with highest diversity will increase the opportunity for mining whole new fitness which might be better than we have.

The idea mechanisms are designed for the purpose we discussed above, and the result is listed in Table II. These rules have been tested 10 replications by an instance of TSP which is KroA100, generations are 100,000, crossover rate is 0.85, mutation rate is 0.05, and size of archive equals population size which is set on 100.

TABLE II
CHROMOSOME SURVIVAL CRITERION

Instance：KroA100				
Removing rule	Average	STD	Improving rate %	Best
Lowest fitness	34122.2	2365.234	8.40%	31885.1
Lowest diversity	32040.05	2463.075	13.99%	28881

From the result of experiments shown in Table 1, the best criterion for removing rule is the lowest diversity, and apply this criterion for lowest diversity, we experiment another factor of archive that is the size of archive. We would like to test if we gather more diversified chromosomes, whether the archive could generate suitable artificial chromosomes for replacement. Therefore, we did several experiments for the sizes of archive are 50, 100, 150 and 200. The main purpose for this experiment is to observe the influence of the factor and the effect degree for different sizes of Archive. The experiment result is listed as Table III.

TABLE III
ARCHIVE SIZE DESIGN EXPERIMENT

Instance：KroA100				
Size of Archive	Average	STD	Improving rate %	Best
50	34100.34	2123.191	8.46%	31099.8
100	32040.05	2463.075	13.99%	28881
150	33424.9	3224.535	10.28%	27392.7
200	32766.62	2706.079	12.04%	29270.2

The purpose of this paper is to improve the searching space for seeking a better fitness than the SGA by applying injecting diversified artificial chromosome. Therefore, the experiment we just discussed about is the cage design in Archive, the following we will propose a philosophy for

injecting the population diversity via artificial chromosomes; we consider this idea via dynamic adjusting fitness in genetic algorithms.

The fitness function is an index for measuring the adaptability for a chromosome solving a problem, and a population evolves generation by generation by way of selecting some chromosomes with excellent fitness. Through this process, it may mislead a population into sub-optimum. In order to inject population diversity, we design a qualification criterion to inject. If population diversity is higher than threshold then it is unnecessary to change the fitness function, else it will trigger the fitness adjustment. As the result we experimented which is shown in Figure 5, even though the diversity is been increased, the fitness searched by the mechanism cannot obtain better fitness.

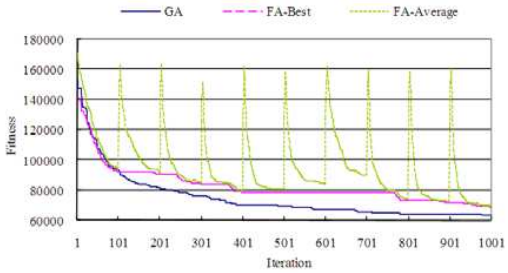

Fig. 5. Dynamic fitness adjustment of promoting population diversity

Moreover, the fitness is usually trapped in repeated convergence conditions; therefore, for the reason to apply this idea, more adjustments should be completed. Besides, the purpose of this paper is to discuss how diversified artificial chromosome influences the quality for searching fitness.

VI. Experimental Results

In order to test the effectiveness of the diversity threshold control mechanism, a set of experimental tests are conducted in the following. First, the experiment is designed to test if the measure of entropy can reflect the diversity of the population during the evolutionary process. Figure 6 indicates that the fitness will be changed when prompt AC is injected.

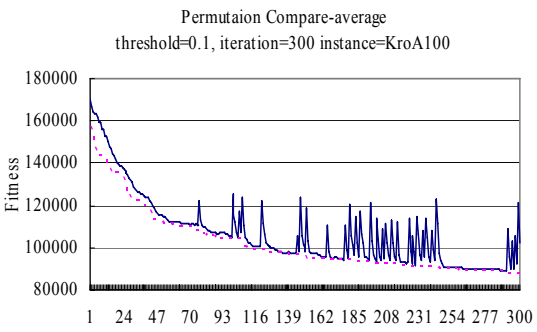

Fig. 6. The effect of injection artificial chromosome

We discover that when the entropy decreases, the diversity also decreases simultaneously. As the result, the entropy will increase whenever we inject AC and return when the diversity decreases. Refer to Figure 7.

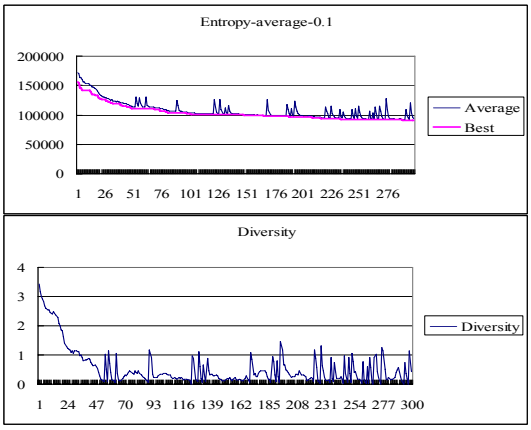

Fig. 7. The population diversity enhance occasion.

Next, we would like to learn how to setup the threshold in order to achieve a satisfactory result. A set of experiments are tested again and the threshold values are setup as 0.01, 0.03, 0.05, 0.07, and 0.09. The final results of these experiments are listed in Table IV.

TABLE IV
Design of Diversity Threshold

Diversity threshold	Fitness	Improvement
0.01	42203.95	-0.68%
0.03	42068.82	-0.36%
0.05	40613.66	3.11%
0.07	41473.86	1.06%
0.09	40663.56	2.99%
GA	41918.49	

Different threshold can cause variation of fitness which can be observed in Figure 8.

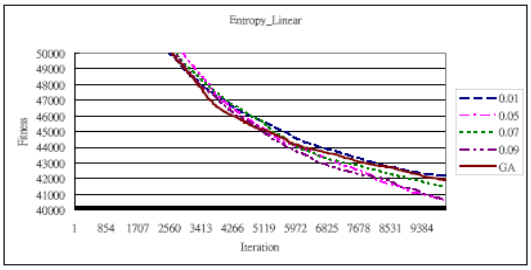

Fig. 8. Fitness Performances for Different Diversity Threshold Values

To further improve the threshold value, some more experiments are conducted again with a large range of diversity threshold values for setting up a suitable threshold and the results are listed in Table V:

TABLE V
Experimental Tests for Different Diversity Threshold

Diversity threshold	Fitness	Improvement
0.1	40004.66	4.57%
0.2	42148.96	-0.55%
0.3	43119.17	-2.86%
0.4	44864.64	-7.03%
0.5	58370.75	-39.25%
GA	41918.49	

From Table V, we can conclude when threshold is 0.1, the performance improvement is the largest, which is shown in Figure 9.

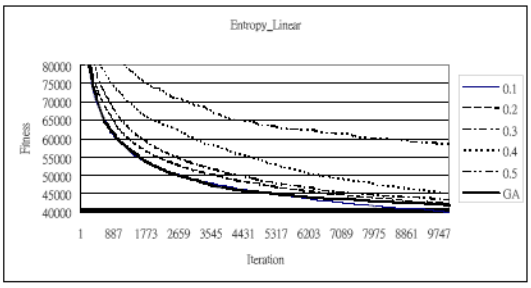

Fig. 9. Fitness Performances in different threshold Values

In these experiments, three instances were chosen for TSPLIB library and computed by SGA and DDCGA. We set the population size as 100, generation as 100,000, crossover probability with 0.85 and mutation probability with 0.05. The Figure 10 shows that when diversity reaches to the threshold, the mechanism will inject AC to enhance the population diversity for searching the un-explorative solution. Therefore, we have a better chance to explore a good solution by way of searching larger space.

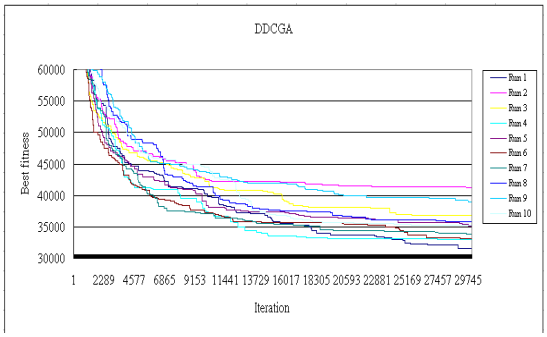

Fig. 10. The convergence of DDCGA

The comparison results of SGA and DDCGA are shown in Table IX. In this research, we found the average fitness of DDCGA is much closer to the best optimal. The improvement is also satisfactory.

TABLE IX
COMPARISONS OF SGA WITH DDCGA IN TSP INSTANCES

	Instance	KroA100	KroA150	KroA200
SGA	Average	37252.93 3	49733.86 7	62336.57 0
	STD	3055.922	2791.933	3746.182
DDCGA	Average	33511.13 0	43053.24 0	57616.45 7
	STD	2886.743	2316.039	3712.752
Improvement	%	10.04%	13.43%	7.57%
Best		29594.00 0	38214.70 0	51969.20 0

VII. CONCLUSION

The experiments revealed a number of interesting features of the DDCGA in relation to combinatorial optimization problem such as TSP. From the phenomenon observed in experiments, the result indicates that the DDCGA is capable of escaping local optima. Second, the control mechanism by injecting artificial chromosomes with high diversity is very effective because as observed from the process the diversity of the population is increased significantly after introducing these artificial chromosomes. The mechanism will re-inject again once the diversity decreases down to a minimum threshold value. Reducing fitness values is highly desirable for real-world applications, because the evaluation is often the time-critical factor in such applications. However, the results showed some variation in the reduction percentages, which indicates that this could be problem dependent. In this research, this control mechanism can simultaneously keep the characteristics of exploration and exploitation which will help traditional GA for extending searching space to obtain better solution.

REFERENCES

[1] T. Black, D.B. Fogel, Z. Michalewicz, and others, "Handbook on Evolutionary Computation," IOP Publishing Ltd and Oxford University Press, 1997.

[2] R.K. Ursem, P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, "Multinational Evolutionary Algorithms," *In Proceedings of the Congress of Evolutionary Computation (CEC-99)*, vol. 3, pp. 1633-1640, 1999.

[3] R. Thomsen, P. Rickers, and T. Krink, "A Religion-Based Spatial Model For Evolutionary Algorithms," *In Parallel Problem Solving from Nature-PPSN VI*, vol. 1, pp. 817-826, 2000.

[4] De Jong, K.A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," PhD thesis, University of Michigan, Ann Arbor, MI,. Dissertation Abstracts International 36(10), 5140B, University Micro_lms Number 76-9381, 1975.

[5] S. Mahfoud, "Crowding and preselection revisited," *Technical Report 92004*, Illinois Genetic Algorithms Laboratory (IlliGAL), 1992.

[6] D.E. Goldberg and J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization," *In Genetic Algorithms and their Applications (ICGA'87)*, pp. 41-49, 1987.

[7] H.G. Cobb and J.F. Grefenstette, "Genetic Algorithms for Tracking Changing Environments," *In: Proceedings of the 5th International Conference on Genetic Algorithms*, Forrest, pp. 523-530, 1993.

[8] R. Thomsen and P. Rickers, "Introducing Spatial Agent-Based Models and Self-Organised Criticality to Evolutionary Algorithms," Master's thesis, University of Aarhus, Denmark, 2000.

[9] G.W. Greenwood, G.B. Fogel and M. Ciobanu, "Emphasizing Extinction in Evolutionary Programming," *In: Proceedings of the Congress of Evolutionary Computation*, Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A. (eds.), vol. 1, pp. 666-671, 1999.

[10] T. Krink, R. Thomsen and P. Rickers, "Applying Self-Organised Criticality to Evolutionary Algorithms," *In: Parallel Problem Solving from Nature-PPSN VI*, vol. 1, pp. 375-384, 2000.

[11] H. Shimodaira, "A Diversity Control Oriented Genetic Algorithm (DCGA): Development and Experimental Results," *In: Proceedings of the Genetic and Evolutionary Computation Conference*, vol 1, pp. 603-611, 1999.

[12] F. Oppacher and M. Wineberg, "The Shifting Balance Genetic Algorithm: Improving the GA in a Dynamic Environment," *In: Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 504-510, 1999.

[13] S. Tsutsui, Y. Fujimoto and A. Ghosh, "Forking Genetic Algorithms: GAs with Search Space Division Schemes," *Evolutionary Computation*, vol. 5, pp. 61-80, 1997.

[14] P. Bak, "How Nature Works," *Copernicus, Springer-Verlag*, New York, 1st edition, 1996.

[15] P. Larrañaga and C.M.H. Kuijpers1 and R.H. Murga1 and I. Inza1 and S. Dizdarevic1, "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators," Artificial Intelligence Review, vol. 13, 1999.