Expert Systems with Applications 37 (2010) 1863–1878

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems

Pei-Chann Chang [a,b,*], Wei-Hsiu Huang [b], Ching-Jung Ting [b]

[a] Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, ROC
[b] Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 32026, Taiwan, ROC

ARTICLE INFO

Keywords:
Artificial chromosomes
Genetic algorithm
Individual diversity
Dynamic diversity

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 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 constantly 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 and to revolve again. A dynamic threshold control mechanism is built up during the evolutionary process to further improve the system performance. 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 other approaches.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

For decades of researches in solving combinatorial optimization problems (COPs), numerous algorithms were proposed for applying to this specific problem. One of these algorithms is named GA (genetic algorithm) which was first presented by Holland. Since then, 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 and Richardson (1987), Goldberg (1989) and Black, Fogel, and Michalewicz (1997).

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.

One of the most successful applications of GAs is found in the area of COPs by Gen and Cheng (1997). They showed that relations between exploration and exploitation seem to be the key to choose the set of correct parameters for a given problem. Some studies considered mixing the probabilistic model and genetic operators together. Chang, Chen, and Liu (2007) and Peña et al. (2004) addressed hybrid algorithms by combining generic chromosomes and artificial chromosomes (ACs) generated by estimation of distribution algorithm together. From the previous philosophy proposed, the ratio of generic chromosomes and artificial chromosomes is determined by a dynamic method. The latter one did not use the probabilistic model until a number of generations were run. Moreover, Chang, Chen, and Fan (2008) proposed evolutionary algorithm with probabilistic models (EAPM) which was not applied generation by generation because the time-complexity is very high when meet sequential problems, therefore, the probabilistic model is used to guide the evolutionary process of crossover and mutation. As a result, the artificial chromosomes were generated with specific intervals. This hybrid style can operate without high computational efforts entailed by generating solutions through probabilistic models.

* Corresponding author. Address: Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, ROC. Tel.: +886 3 4638800x2305.
E-mail address: iepchang@saturn.yzu.edu.tw (P.-C. Chang).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.07.066

As Ursem et al. (1999) pointed out that 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 et al.'s (1999) philosophy with decreasing and increasing diversity operators to control the population diversity. Therefore, refer to Mauldin (1984), 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 then 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 which will maintain the population diversity until the threshold d_{high} is met again. Since the DAC is injected, the population diversity has activation for next evolution.

2. Literature review

The major focus of the research is to apply the AC generation mechanism to generate DACs to be injected into the evolutionary process thus the diversity of the population can be increased up to a pre-defined level. Then a dynamic threshold control mechanism is applied to control the evolutionary process once the diversity drops down to a minimum level the DACs are injected into the population again. The dynamic control mechanism is applied to ensure the EA will not be trapped into local optimum in early stage and thus a multi-convergence process can be detected. Therefore, an overall global optimum can be reached. The problems set tested will be TSP problems. Therefore, the TSP problems are surveyed in Section 2.1. In addition, the diversity control in genetic algorithm is also reviewed in Section 2.2.

2.1. Traveling salesman problem

Through the years the traveling salesman problem has attracted a lot of attentions from academic researchers and industrial practitioners. 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. Larrañaga, Kuijpers1, Murga1, Inza1, and Dizdarevic1 (1999) introduced the TSP problem whose 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.

2.2. Diversity control in genetic algorithm

Amount of researchers announced that diversity measure and control can influence the efficiency of convergence. Various strategies are applied to maintain or increase the population diversity. Nonetheless, a modification on the selection operation which was done by Baker (1989) for the task has received much attention. Moreover, Mori, Yoshida, Tamaki, Kita, and Nishikawa (1995) introduced a notion of thermo dynamical genetic algorithm where the survival of individuals is regulated by means of monitoring the free energy within the population. The modification on the selection operation also be done in the cross generational-sense which proposed by Whitley (1989), Eshelman (1991) and Shimodaira (1996). Whitley (1989) has proposed a GENITOR system where offspring generated by standard operators are chosen for replacing parents based upon the ranks of the individuals. In contrast to Whitley (1989) and Eshelman (1991) recommends the application of mating restriction while Shimodaira (1996) suggests the use of variable-rate mutation as a means to create offspring. Then a cross-generational survival selection is carried out using a standard fitness-based selection technique in both cases.

In addition to the early works described above, another genetic algorithm has been specifically developed by Shimodaira (1997) to handle the issue of population diversity; this algorithm is called a diversity control oriented genetic algorithm or DCGA. Similar to most genetic algorithms, offspring in the DCGA are generated using standard crossover and mutation operators. However, during the cross-generational survival selection, duplicated individuals in the merged population containing both parent and offspring individuals are first eliminated. Shimodaira (1999) further proposed that the remaining individuals are selected based on either the associate fitness or the consideration on both the fitness and the genomic similarity between the interested individual and the elite individual. The performance of the DCGA has been benchmarked using various single objective test problems in the paper proposed by Shimodaira (2001). With a minor modification the DCGA can also be used in multi-objective optimization. One possible approach for achieving this is to integrate the DCGA with other genetic algorithms that are specifically designed for multi-objective optimization such as a multi-objective genetic algorithm or MOGA which was proposed by Fonseca and Fleming (1993, 1995, 1998). This implies that the modified version of the DCGA, which can be referred to as multi-objective diversity control oriented genetic algorithm or MODCGA, can also be benchmarked using multi-objective test problems.

In contrast to DCGA, this research takes a step further by constantly 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. A dynamic threshold control mechanism is built up during the evolutionary process to further improve the system performance.

3. A dynamic diversity control genetic algorithm

Population diversity is a key issue in the performance of evolutionary algorithms. Therefore, Weerayuth and Chaiyaratana (2002) proposed that 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 controlling chromosome diversity. 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.

In Fig. 1, there are totally two phases in the process of dynamic diversity control genetic algorithms (DDCGA); Phase one is SGA, which is shown on the right hand side of the figure. When the population diversity is higher than the pre-defined threshold, the procedure is the same as the traditional GA approach. Phase two is a dynamic artificial chromosomes mechanism (DAC), which is shown on the left hand side of the figure. DAC will enhance the population diversity by injecting diversified artificial chromosomes into the mating pool. The architecture of a DDCGA is shown in Fig. 1. Through the dynamic control of population diversity, DDCGA will inject the DAC into the population to prevent premature convergence and extend the searching spaces which are unexplored in the traditional GA for locating a better solution.

As 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. DDCGA will trigger an artificial chromosome control mechanism to generate a set of artificial chromosomes with high diversity and inject these chromosomes into the population. Although, the fitness of these artificial

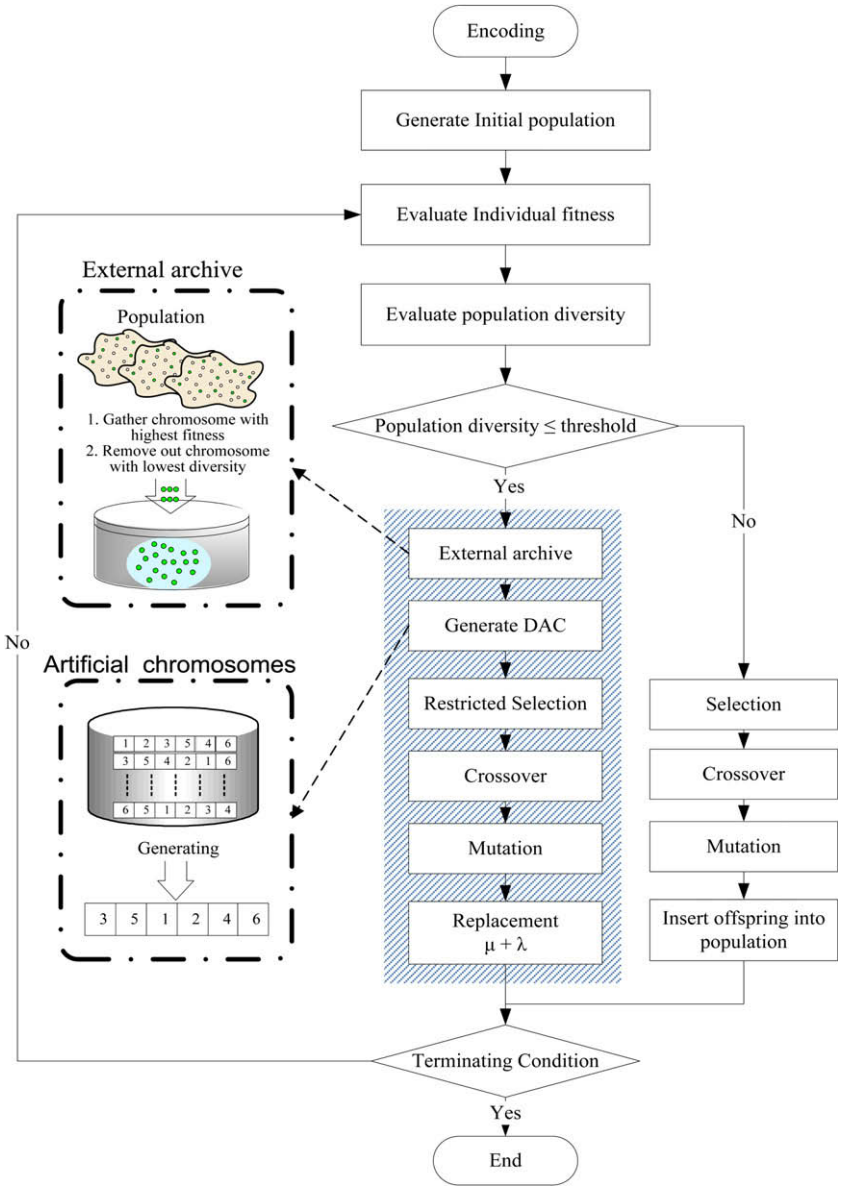

Fig. 1. The architecture of DDCGA.

chromosomes may not as good as those within the original population, however, the diversity of these ACs is pretty high. Hopefully, the exploration of the evolutionary process can restart again. In the ultimate goal, a near-optimal chromosome with good fitness can be generated in the long run. The pseudo code of the DDCGA algorithm is listed as follows:

Population: The sample used in the genetic algorithm for searching solution

Termination Condition: The generations' number is reached

```
Initialize Population
while termination condition is not satisfied do
  Evaluate each fitness and diversities of population
    if population diversity > threshold then
      Selection operator
        Crossover operator
        Mutation operator
    else
        Create artificial chromosomes
        Select parents and artificial chromosomes
        Crossover operator
        Mutation operator
    μ + λ Replacement
  end if
end while
```

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 chromosomes generated during 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.

Finally, the idea of multiple archives will be embedded in DDCGA and the new model is named as MDDCGA to discriminate the difference between original DDCGA and the one with multiple archives. The basic concept is that by introducing multiple archives into the system then the artificial chromosomes generated will have a variety of sources. The diversity will be more significant when compared with the single archive. The architecture of MDDCGA is shown as in Fig. 2, after generating initial population, we instantly generate multiple external archives for chromosomes with better fitness, whenever the amount of chromosomes in any one of the external archives reaches the pre-defined limit. The system will proceed to remove the chromosomes with lowest diversity.

4. An artificial chromosome generation mechanism

As discussed in the above section, the quality and variety of AC will influence the convergence of the population. Therefore the facilitated AC mechanism is extraordinarily important for GA to extend a 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 collect chromosomes generated in previous generations with high diversity and fairly good fitness measure into the external archive and use the gene information in the archive to generate artificial chromosomes with good diversity. At the beginning, it collects chromosomes whose fitness is the current best for the first couples of iterations. When the archive size equal to the population size, then it gathers a chromosome with the current best into the archive and removes a chromosome with the lowest diversity in the archive.

All the gene information in the archive will be converted into a dominance matrix, by calculating the total number of genes for each specific gene in each position. This dominance matrix then will be applied to generate artificial chromosome. The assignment sequence for each position is assigned randomly. Thereupon, we will assign one gene at a time to each position by the roulette wheel selection method. Based on the probability of each gene appearing in each position, the appearing frequency will be derived from the dominance matrix. After we assign one gene to a position, the specific gene and position in the dominance matrix will be removed. Then, the process repeats to select the next gene to next position until all genes are assigned to a fixed position.

Assume there is a five city TSP problem and the population size is set to 10, which means an archive will gather 10 chromosomes from previous evolutionary processes. Moreover, we convert the gene information from these 10 chromosomes into a dominance matrix. For 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 in each position as shown in Fig. 3.

The appearing frequency of each gene is recorded to the corresponding position as shown in the dominance matrix. 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.

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

Based on the above dominance matrix in Fig. 3, a probability matrix is derived. Then, we randomly assign the first gene and assume it is at position 3. The frequency of each gene at position 3 is (1, 3, 1, 1, 4) for gene 1–5. There are 10 chromosomes, i.e., $N = 10$. The corresponding probability for gene 1 is 1/10; gene 2 is 3/10, and so on. Hence, we accumulate the probability from gene 1 to 5 and utilize the roulette wheel selection to assign a gene to each position as shown in Fig. 4.

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 the archive. Artificial chromosomes are generated and injected into the mating pool, and then a tournament selection procedure will select one chromosome from regular chromosomes and the other from artificial chromosomes for crossover and mutation. Afterward, chromosomes with better fitness evaluation will be reserved in the archive. Consequently, chromosomes with higher diversity are preserved to the next generation.

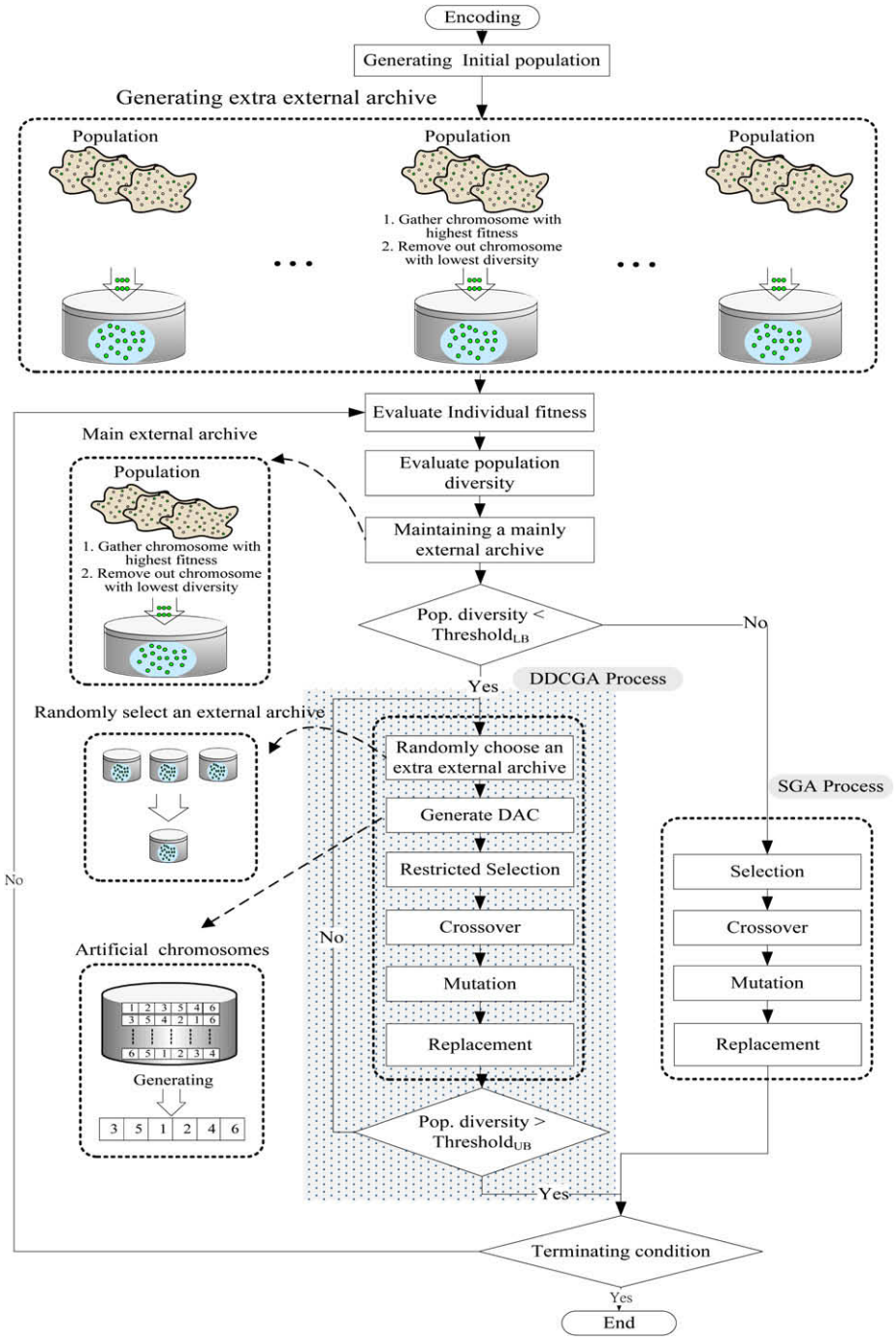

Fig. 2. The architecture of MDDCGA.

5. Diversity measures and multiple-archive design

To provide a better exploring capability, a good archive design which can generate AC with high diversity is a key in this research. Therefore, the effectiveness of archive design will influence the convergence speed and final quality of the fitness of the population. This research proposes a new concept of dynamic diversity control within the evolutionary process; 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 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 applied such a measure 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 inject 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, owing to the selection pressure the evolutionary processes will gradually reduce the diversity level again. However, the control mechanism will ensure the artificial chromosomes will be re-introduced

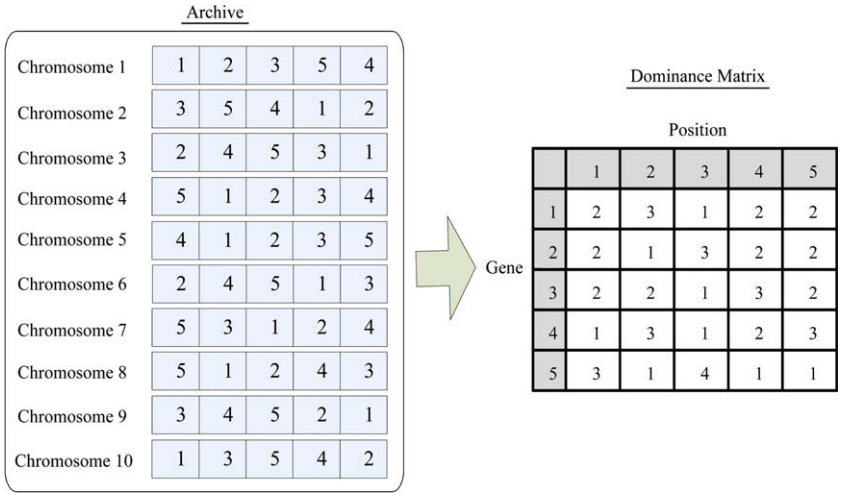

Fig. 3. The transferring process of dominance matrix.

Gene	Position 3	Probability	Accumulation
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. 4. The probability matrix from dominance matrix transformation.

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 section, 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.

5.1. Diversity of an individual chromosome

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.

In this research, the diversity of the population during the evolutionary process is set according to the following diversity-measure:

$$diversity(P) = \frac{1}{|L||P|}\sum_{i=1}^{|P|}\sqrt{\sum_{j=1}^{N}(s_{ij}-\bar{s}_j)^2} \tag{3}$$

where $|L|$ is the length of the diagonal in the search space, P is the population, $|P|$ is the population size, N is the dimensionality of the problem, s_{ij} is the jth value of the ith individual, and $\bar{s}_j$ is the jth value of the average point s.

The DDCGA 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 DDCGA 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 DDCGA should be able to escape local optima because the operators will enforce the population with 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.

5.2. Population diversity measure

In different kinds of combinatorial problems, diversity measures are tightly problems depended. There are two different ways to measure individual diversity in combinatorial problems. One is measure the difference between two genotypes while the other is a structural difference measure based on mathematic 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. There are four different diversity measure approaches proposed in the literature and they are listed as follows.

5.2.1. 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:

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

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

$$\text{where } M = \begin{cases} L^2 - 1/2, & \text{if } L \text{ is odd.} \\ L^2/2, & \text{if } L \text{ is even.} \end{cases} \tag{5}$$

5.2.2. By Euclidean distance

Euclidean distance is used to a real encoding; the concept is the same with hamming distance in permutation encoding

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

5.2.3. 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

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

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

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

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

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

5.2.4. 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 ith locus ($i = 1\cdots n$) is defined as follows:

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

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

However, 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

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

where $\bar{d}$ is the average diversity, $d_{\max}$ is the maximum diversity and $d_{\min}$ is the minimum diversity of the archive.

5.3. Main archive design

To set up the best parameter combinations, we firstly list the influence coefficient to be the factors for analysis in Table 1. 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 1
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 allele. 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 Fig. 5.

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 ideal mechanisms are designed for the purpose we discussed above, and the result is listed in Table 2. 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.

From the result of experiments shown in Table 2, 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

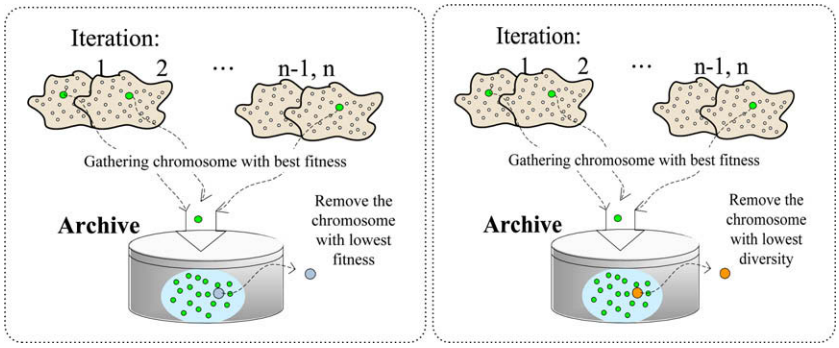

Fig. 5. Illustration of archive infrastructure.

Table 2
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

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 3.

5.4. A multiple-archive design

Base on the above result, the population diversity can be enhanced after DACs are introduced; however, the improvement is not significant in generating qualified DACs. Tsutsui, Fujimoto, and Ghosh (1997) proposed searching space division, Potter and DeJong (1994), Pimpawat and Chaiyaratana (2001) and Boonlong, Chaiyaratana, and Kuntanapreeda (2002a, 2002b) all proposed co-evolutionary strategy for enhancing GA, which inspires us to design multiple-archive to further extend the diversity of the artificial chromosomes generating by breeding more qualified chromosomes in different archives.

The multiple archives are generated by gathering qualified chromosomes from different seeds of SGA process. A case example is demonstrated in Fig. 6 and there are totally eight archives generated from SGA with different initial seeds. These different archives just like seed chromosomes bred in different cages and the system will inject ACs from different archive each time the injection process is triggered. This process will improve the diversity of the artificial chromosomes generated from these archives. There are more chances as shown in Fig. 6 for these eight archives to converge into a global optimum.

All extra-archive have been collected, those are called multi-archive. Not only the population evolved, but also the main archive evolved too with the main archive collecting rule. It will randomly choose an extra-archive from multi-archive as the main archive. Now, we have enough material of different explored space to generate DAC. While population diversity is lower than the threshold low, the DDCGA process is triggered and it will randomly choose an archive from multi-archive to generate DAC until population diversity is higher than threshold high. Those will increase population diversity and make evolution sustained.

Table 3
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

5.5. Design of parameters

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, we focus on discussing about the cage design in archive; we propose a philosophy for injecting the population diversity via artificial chromosomes and dynamic injecting DAC for fitness in genetic algorithms. In common GA 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 shown in Fig. 7, even though the diversity is been increased, the fitness searched by the mechanism cannot obtain better fitness. Moreover, the fitness is usually trapped in repeated convergence conditions; therefore, for the reason to apply this idea, more adjustments should be completed.

In order to test the effectiveness of the diversity threshold control mechanism, a set of experimental tests are conducted. First, the experiment is designed to test if the measure of entropy can reflect the diversity of the population during the evolutionary process. Secondly, the evolving processes will re-convergence in SGA for better fitness, which the fitness should be better as we expect. Fig. 8 indicates that the fitness will be changed when prompt DACs are injected, and the process also re-convergences after injecting DACs.

In Fig. 9, we have discovered that when the entropy decreases, the diversity also decreases simultaneously. As the result, the entropy will increase whenever we inject DAC and return when the diversity decreases.

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 4, the performances are all positive when the threshold over than 0.05, nevertheless, the performance of the result obtained is the most significant when the diversity threshold reaches 0.05.

From the observation in Fig. 10, different threshold design can cause variation of fitness which is shown in Fig. 10. In this research, we firstly apply whose interval is 0.2, from 0.01 to 0.09, for testing

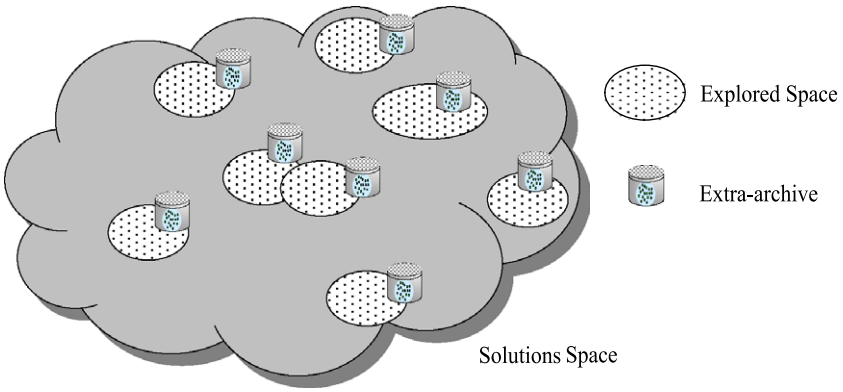

Fig. 6. A case example for 8-archive design.

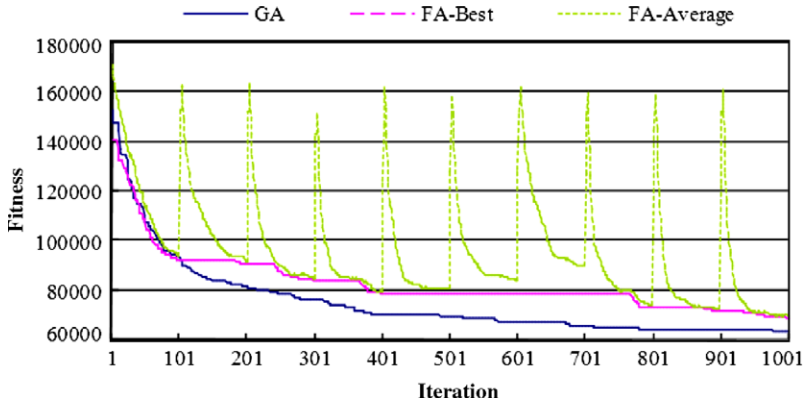

Fig. 7. Dynamic fitness adjustment of promoting population diversity.

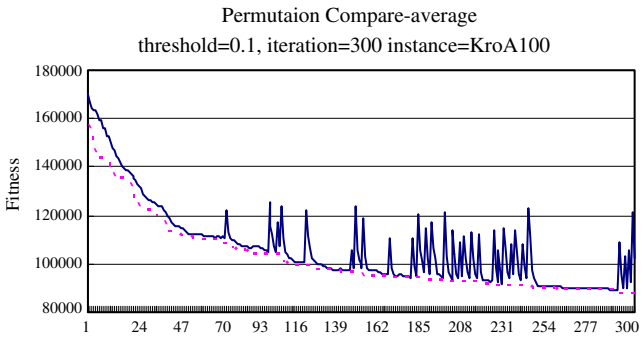

Fig. 8. The effect of injection artificial chromosome.

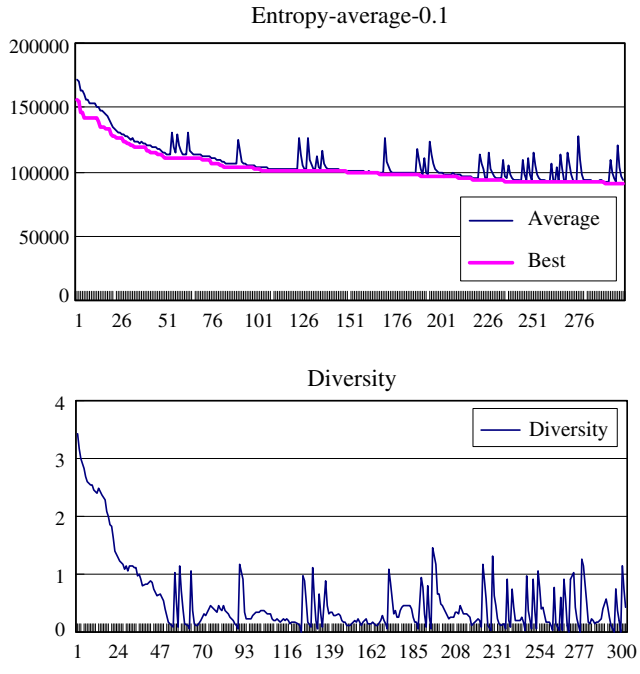

Fig. 9. The population diversity enhance occasion.

the sensitivity; the major purpose is to find the best parameter to decide the threshold.

To further improve the threshold value, 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 5.

From Table 5, we can conclude when threshold is 0.1, the performance improvement is the largest, which is shown in Fig. 11.

In these experiments, three instances were chosen from 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.

Fig. 12 represents 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 better chances to explore a good solution by way of searching wider space.

The process for injecting DAC is infinite; therefore, the DAC will be injected without terminating. Nevertheless, the overhigh

Table 4
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	

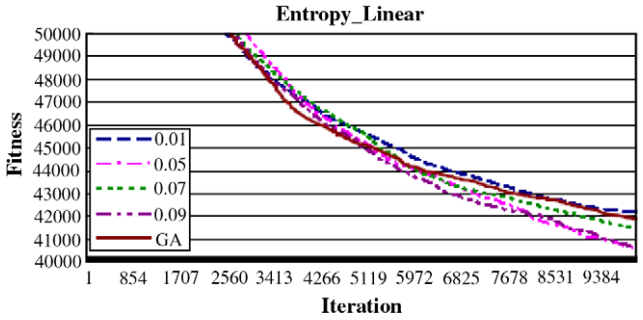

Fig. 10. Fitness performances for different diversity threshold values.

Table 5
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	

diversity will cause SGA process hardly converges; we therefore need additional threshold to prevent the diversity overhigh. The threshold here is named upper threshold, whose experimental result is shown in Fig. 13.

In the prior experiment for deciding the upper threshold, each gap for increasement is defined as 0.05 for analysis about the diversification. We clearly discover that the mechanism is effective when the interval is between 0.2 and 0.6.

In Fig. 14, experiment for increasing gap from interval 0.2 to 0.9 is assigned as 0.1 for observing the difference. For improving the efficiency of DDCGA, we will introduce continuous probability density function (CPDF).

In Fig. 15, the experiment is design for different AC gathering strategy, which is continuous probability density function, this means, dominance matrix will continuously gather genes information without restriction for terminating.

6. Dynamic threshold control design

From the previous experiment for upper threshold, we confirm that the threshold for restricting chromosomes with overhigh diversity is essential but with difference peak performance at different upper threshold. Therefore, we consider proposing a dynamic mechanism for controlling the threshold. This concept of dynamic threshold control is to inject AC when the diversity difference is steady over certain iterations and terminate when it reach to the upper threshold, the measuring and evaluating the diversity of population for judging the time for injecting DAC is described as follows:

1. Measuring the diversity of the population.
2. If the diversity value stored in achieve is below threshold value d_{low}, the diversified artificial chromosomes will be injected into the population for increasing the diversity of population.

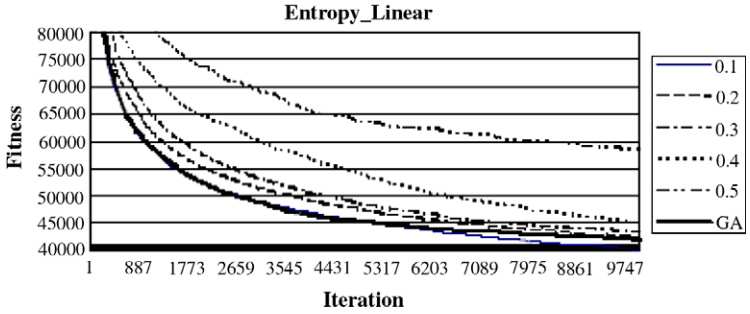

Fig. 11. Fitness performances in different threshold values.

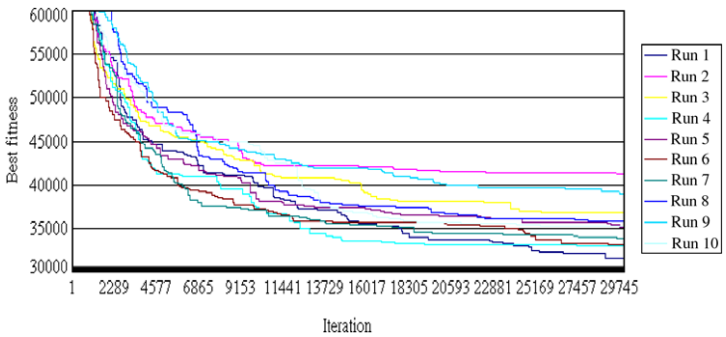

Fig. 12. The convergence of DDCGA.

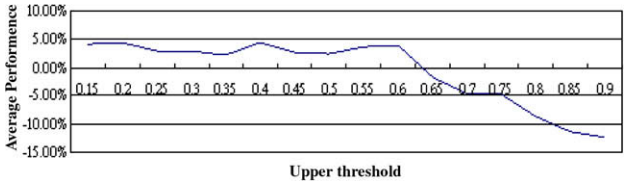

Fig. 13. Analysis of different d_{high} threshold values in DDCGA.

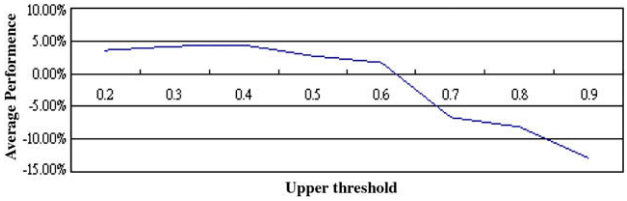

Fig. 14. Analysis of different d_{high} threshold values in MDDCGA.

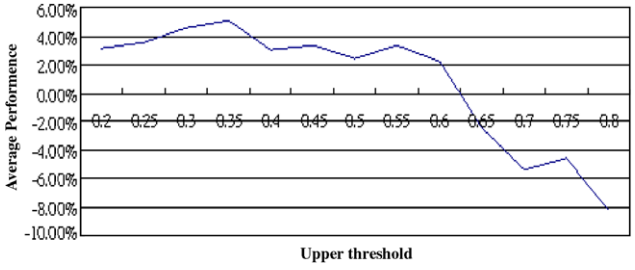

Fig. 15. Analysis of different d_{high} threshold values in DDCGA with CPDF.

3. Adjusting the selection pressure to maintain the diversity of the population in high level for a time period, i.e., T (avoid sudden drop again in diversity).
4. Return to the SGA process. If $P = P_{max}$ then stop; otherwise go to step 1.

For updating the proposed approach and simultaneously keep chromosomes with high diversity, we apply the chromosomes with attribute of diversity, which are d_{low} and d_{high}, to control the injection mechanism of artificial chromosome. This mechanism actives to inject artificial chromosomes when chromosome diversity value reaches to d_{low} which is set to 0.1 that was applied to be the threshold in previous experiment in this research; stops to inject when reaches to d_{high} which is set from 0.1 to 0.9. For maintaining the robust of this developed approach, and verify the stability through comparing with SGA, we apply the instances from TSPLIB which are KroA100, KroA150 and KroA200 to decide the parameter of threshold.

Table 6 shows the experimental result for applying instance KroA100 with d_{high} value from 0.1 to 0.9, and with 10,000 iterations. From the experiment result in Table 6, the best solution is very close to the solution of SGA.

In Fig. 16, the improvements for each value of d_{high} are all close to the horizontal axis of 0.00%, which is the optimal solution of SGA. The plotted lines represent that the DDCGA outperforms better than SGA when line segments are above the axis, which means, the positive value indicates the DDCGA has improvement comparing with SGA. The improvements are positive when the upper threshold is between 0.6 and 0.8, the most significant improvement happen when the upper threshold is set to 0.8. The improvement is rapidly decrease after upper threshold is 0.8, which means, the overhigh diversity will influence the fitness, and this experiment also gives us provident idea for controlling the chromosome diversity.

Table 6
Experiment with different d_{high} value for KroA100 (interval: 10^{-1}).

d_{high} value	Average	Best	STD	Performance (%)
0.1	43093.46	38150.80	2785.83	−0.01
0.2	43425.26	41130.50	1893.11	−0.78
0.3	43907.99	39719.10	2619.10	−1.90
0.4	42992.81	39657.10	2024.76	0.22
0.5	44127.83	39950.40	2685.06	−2.41
0.6	37337.20	37337.20	3684.24	0.53
0.7	42667.23	38189.00	3448.90	0.97
0.8	42590.42	39251.30	2609.55	1.15
0.9	47856.39	43790.70	2479.44	−11.07
SGA	43087.29	39392.90	3368.63	

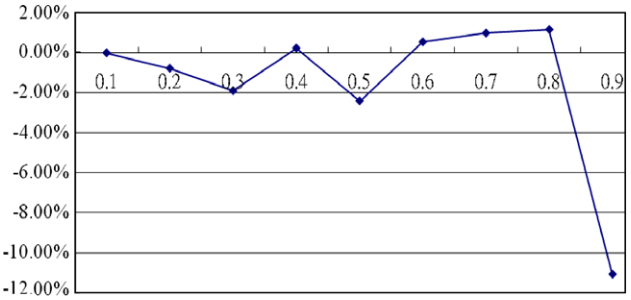

Fig. 16. The improvement of DDCGA comparing with SGA for KroA100 (interval: 10^{-1}).

To further improve the efficiency of DDCGA, we estimated more iteration will help DDCGA to derive better solutions; hence, we increased the iterations up to 50,000. Moreover, from the experiments we proposed previously, we found that the solution is better when d_{high} value is between 0.6 and 0.7. We firstly try to find the proper threshold for d_{high}, and then we transformed the intervals of d_{high} (0.1, 0.9, 10^{-1}) into d_{high} (0.5, 0.9, 10^{-1}).

The paper here lists the data of d_{high} from 0.6 to 0.8 in Table 7 since the performances are all positive, and it will take too much space if we want to enlist the whole range of d_{high}., the performance of DDCGA is much better than SGA and the improvement rates are almost over 10%, which means, the diversity is well-controlled when the upper threshold is between 0.6 and 0.8. Moreover,

Table 7
Experiment with different d_{high} value for KroA100 (interval: 10^{-2}).

d_{high} value	Average	Best	STD	Performance (%)
0.61	34118.90	29170.60	2468.66	12.30
0.62	34085.17	29385.70	2682.93	12.39
0.63	35041.88	30674.00	1991.77	9.93
0.64	34255.34	30630.70	2613.04	11.95
0.65	34070.36	31023.10	2260.44	12.43
0.66	34306.05	29990.20	2855.45	11.82
0.67	34821.08	32164.00	2761.20	10.50
0.68	32901.40	30909.90	1692.12	15.43
0.69	34503.41	30031.40	3093.21	11.32
0.7	33329.46	30290.20	2003.61	14.33
0.71	32893.80	28752.30	2870.65	15.45
0.72	33293.23	29947.80	2075.87	14.43
0.73	34048.72	31723.50	1625.35	12.48
0.74	34884.82	31216.60	2926.88	10.33
0.75	33078.93	30780.80	2013.14	14.98
0.76	34147.61	30431.20	2963.21	12.23
0.77	33576.24	30911.70	1773.29	13.70
0.78	34764.11	30447.10	2479.47	10.65
0.79	34431.70	30250.30	2851.80	11.50
GA	38905.67	33841.20	3179.22	

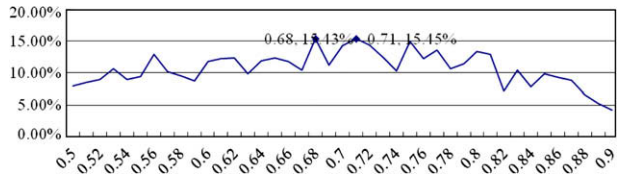

Fig. 17. The improvement of DDCGA comparing with SGA for KroA100 (interval: 10^{-2}).

Table 8
Experimental results of KroA100.

Methodology	Population size	CR	MR	LT	UT	AS	EA	CI
SGA	100	0.9	0.3	–	–	–	–	–
MDDCGA	100	0.9	0.3	0.1	0.7	100	7	1000
DDCGA	100	0.9	0.3	0.1	0.7	–	–	–

Table 9
Experimental results of KroA100.

Instance: KroA100				
Algorithm	Improvement	Average	Best	STD
SGA	–	27117.92	25347.5	1260.39
MDDCGA	2.04%	26564.60	25093.1	1181.08
DDCGA	5.08%	25739.63	23924.1	710.59

when d_{high} reached 0.71, the performance even over than 15%, the results are plotted in Fig. 17 with total experimental data.

In Fig. 17, we can easily discover the situation of improvement is very large, and when we refer to the data in Table 7, we will also find the STD of DDCGA is much less than SGA's. Therefore, the efficiency of DDCGA is much better than SGA's, but we still have to do more experiments for verifying, we therefore test by different instances.

Based on the result of the previous experiment, DDCGA is verified that the philosophy is effective and with the characteristic of low variance. After each injection, the average fitness of the population will be raised (worse) than the current average fitness since chromosomes with higher diversity are merged into the population. After the diversity of the current population is increased up to certain level, the population with higher diversity will continue to evolve under the standard process of GA. The evolving process after injecting the diversified artificial chromosomes into the population is so-called "Re-convergence". Rely on the process of Re-convergence, the fitness will occasionally obtain a better fitness in the extended solution spaces and will improve the original defects for insufficient searching space. Base on the experimental results, the parameters are decided as Table 8, where the crossover rate is CR, mutation rate is MR, lower threshold is LT, upper threshold is UT, size of archive is AS, external archives is EA and the initial external archive with fixed collecting iterations is CI.

7. Experimental results

In this section, we did several experiments for testing complexity of problems; therefore, we take several different numbers of cities which are KroA100, KroA150, KroA200, PR299 and PCB problem as test instances from TSPLIB. Due to the significant experimental result of parameters combination of DDCGA, all the following experiments of each parameter are adopted by the above parameters combination, replicated for 20 times, and the DDCGA with archive restricted in size 100 will be shown in the following experiments.

In the result of KroA100 in Table 9, the performances of the hybrid algorithms DDCGA and MDDCGA are significant, this verify that the proposed dynamic threshold control is effective. The result also indicates that the DDCGA is more effective than MDDCGA, and DDCGA is more stable than MDDCGA. The convergence of KroA100 is shown as Fig. 18 which shows the result that the DDCGA is effective than other approaches in this research.

Table 10
Experimental results of KroA150.

Instance: KroA150				
Algorithm	Improvement	Average	Best	STD
SGA	–	33495.67	31194.2	1440.86
MDDCGA	1.09%	33130.30	30338.7	2199.20
DDCGA	2.27%	32736.97	30700.1	1287.89

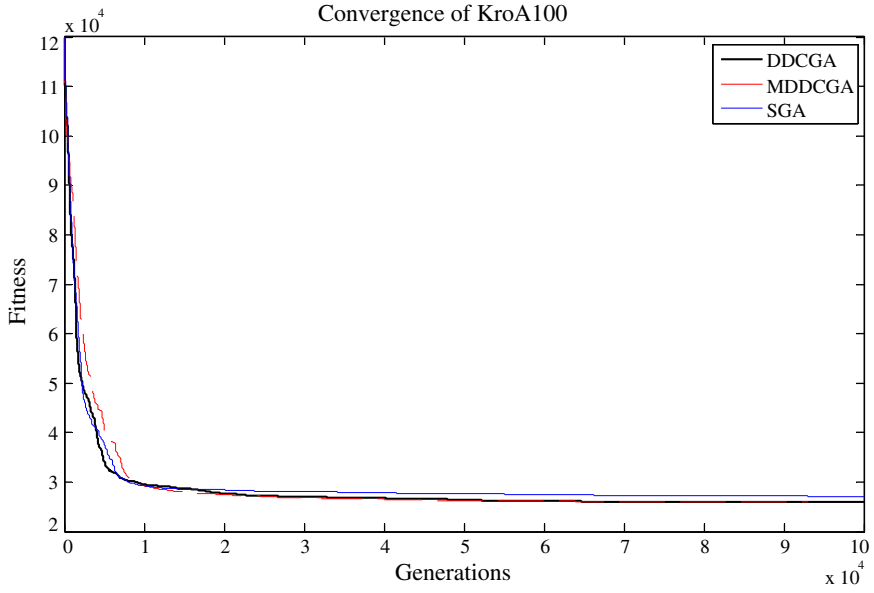

Fig. 18. Convergence of KroA100.

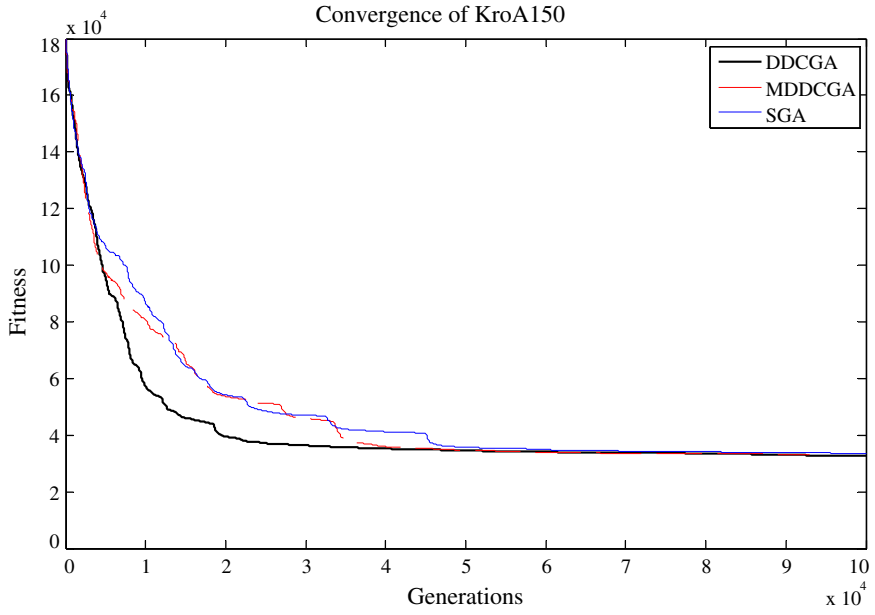

Fig. 19. Convergence of KroA150.

Table 11
Experimental results of KroA200.

Instance: KroA200				
Algorithm	Improvement	Average	Best	STD
SGA	–	50296.29	38755.9	34374.52
MDDCGA	−20.43%	60571.12	38499.3	41519.93
DDCGA	16.12%	42189.31	37786.9	2666.36

Table 12
Experimental results of PR299.

Instance: PR299				
Algorithm	Improvement	Average	Best	STD
SGA	–	207418.58	69601.6	158949.90
MDDCGA	−28.41%	266349.01	77725.3	152190.25
DDCGA	18.32%	169421.01	78832.6	132580.45

The results of KroA150 are shown in Table 10 which indicated the same phenomenon as KroA100 experimental result, which means the performance of DDCGA is better than the others and the convergence occurs after 7000 generations. From the results in Fig. 18, the convergence time of DDCGA is much earlier than other algorithms, and the average is also lower than others.

As we mentioned in previous section, due to the establishment of probability matrix, the more data are gathering, the dominance matrix will be much better than those without these data which are shown in Fig. 19.

Although MDDCGA is constructed from DDCGA, however the results showed that multiple archives cannot enhance the

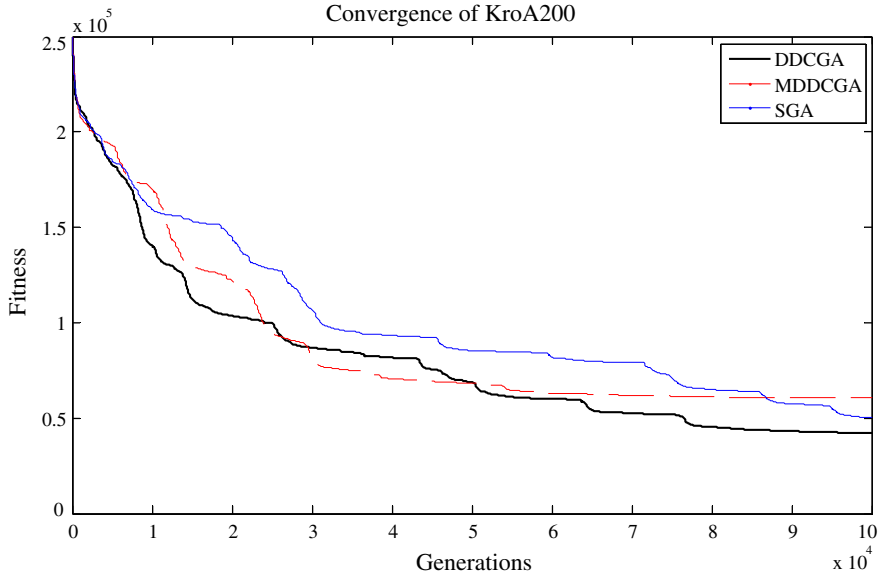

Fig. 20. Convergence of KroA200.

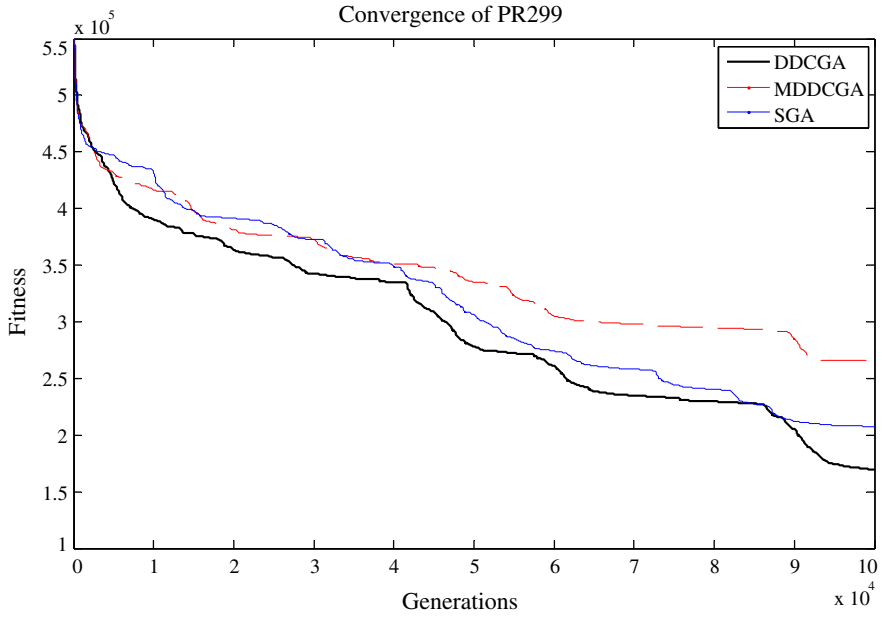

Fig. 21. Convergence of PR299.

Table 13
Experimental results of PR299 with CR0.85 MR0.05.

Instance: PCB442				
Algorithm	Improvement	Average	Best	STD
SGA	–	118385.6	111637	6216.974
MDDCGA	20.68%	93907.77	85131	7423.888
DDCGA	23.46%	90611.29	83692.4	4324.051

Table 14
Experimental results of PCB442 with CR0.85 MR0.05.

Instance: PCB442				
Algorithm	Improvement	Average	Best	STD
SGA	–	136426.3	125779	7768.42
MDDCGA	17.72%	112253.2	104121	4303.779
DDCGA	18.46%	111235.2	103740	4820.588

exploration process. The result is shown as Table 11 and the convergence of KroA200 is plotted as Fig. 20.

In the result of testing instance KroA200, we obtained the similar conclusion, which is, the DDCGA outperforms than other approaches proposed in this research, which is shown in Fig. 20.

The results in Table 12 show that only DDCGA can enhance the exploration of SGA and it also has satisfactory performance in this instance. The convergence diagram of PR299 is shown as Fig. 21.

The results in Table 12 show that the approach in this research could not surpass the SGA in convergence which is also plotted in

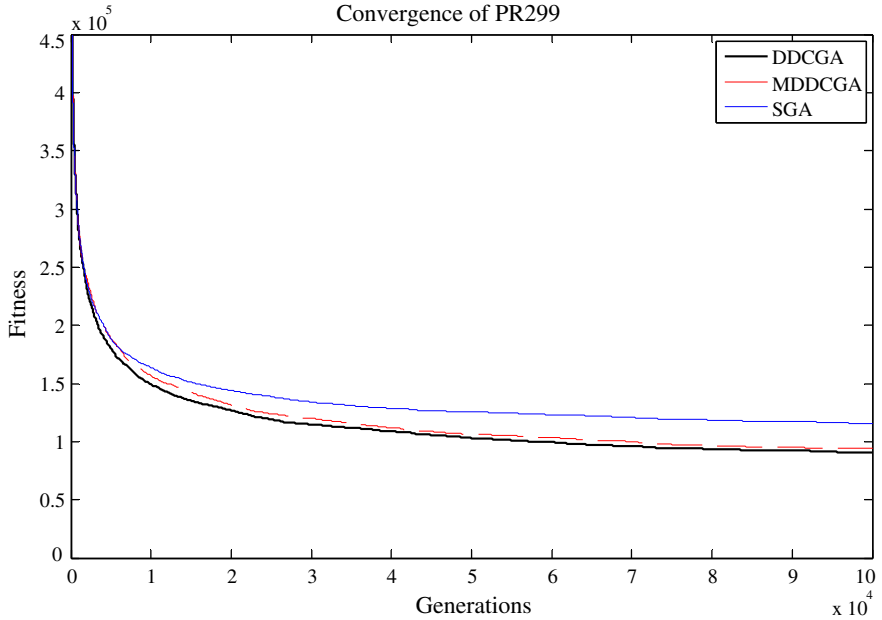

Fig. 22. Convergence of PR299 with CR0.85 MR0.05.

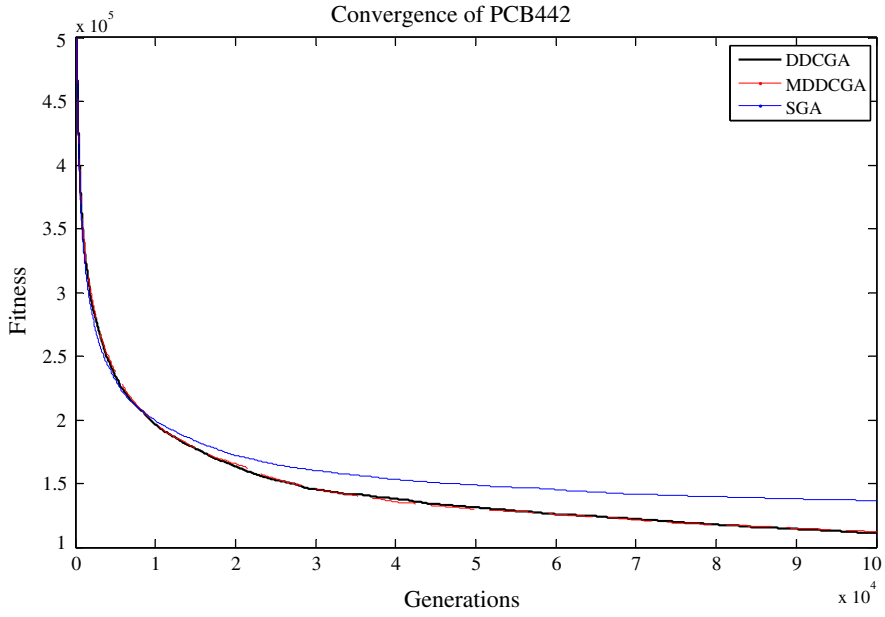

Fig. 23. Convergence of PCB442 with CR0.85 MR0.05.

Fig. 21; however, via monitoring the population diversity shows that SGA cannot exploit anymore. The mutation rate is still high that causes SGA just have exploration but without exploitation in lager-size instance of TSP and it also cannot trigger off the lower threshold. In another word, the efficiency of our approach is just similar to SGA; we therefore considered that the crossover and the mutation rate should be redesigned for improving the problem of un-exploration to overcome the problem, after all, DAC is injected when the SGA stops convergence. In the test instance of PCB442, the best parameter combination for the crossover rate is 0.85 and the mutation rate is 0.05. This combination is applied in SGA for obtaining better fitness; we therefore can inject DAC for re-convergence.

After we redesigned the parameter combination, we adopt the combination to test the above instance for better results. The result in Table 13 shows that the new parameters combination increases the performance of the approach proposed in this research, the DDCGA, even the MDDCGA all outperform than before.

From Fig. 22, the proposed approaches are much more effective than SGA. Once SGA starts to converge into a local optimum, the proposed approaches would start to inject DAC to further improve the searching space. Therefore, the proposed approaches can enhance SGA to re-converge, this mechanism cause SGA to extend the searching space where are not searched in previous evolving process. That is why DDCGA can converge into a better solution than SGA.

A larger instance with 442 jobs is applied to test the performance of DDCGA and we named the instance as PCB442. The parameter combinations have been redesigned, the experimental results are shown in Table 14. The proposed approaches can enhance the exploration ability of SGA and it also has a very good performance in this instance testing. Moreover, from Fig. 23, we also can discover the similar result as in instance PR299, which is, the proposed approaches can outperform than SGA significantly.

Consequently, in Fig. 23, the proposed approaches with similar archive strategies also outperform other previous approaches without optimal parameter combination. From iteration 2 K, when SGA stops to converge, DDCGA starts to take over by injecting DAC to re-converge, therefore, the result of proposed approached can outperform SGA approach.

8. Conclusion and future works

Firstly, the experiments conducted in this research revealed a number of interesting features of the DDCGA in solving the combinatorial optimization problem such as TSP. From the phenomenon observed in experiments, the result indicates that the DDCGA is capable of escaping local optima and further extending its searching spaces. Secondly, 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 drops down to a minimum threshold value. Locating a near-optimal fitness values is highly desirable for real-world applications, because the evaluation is often a time-critical factor in such applications.

However, the results showed some variation in the percentage improvement of the fitness value, which indicates that this percentage of improvement could be problem dependent. In summary, the control mechanism of DDCGA can simultaneously keep the characteristics of exploration and exploitation which will help traditional SGA in extending searching space to obtain better solutions. In this paper, several approaches are proposed for extending searching space, experimental results show that our approaches are very effective respectively.

We here conclude that our approach with DAC is effective when solving the complex discrete combinatorial problems which SGA will be trapped in local optimum. Especially, when the complexity of problem is exponentially increasing, we therefore propose an algorithm with DAC which can enhance SGA to further explore unsearched space. In the future, if the clustering techniques can be applied in the multiple-archive designs, the extended searching space can be further improved.

References

Baker, J. E. (1989). *An analysis of the effects of selection in genetic algorithms*. PhD thesis, Computer Science Department, Vanderbilt University, Nashville, TN.

Black, T., Fogel, D. B., & Michalewicz, Z. (1997). *Handbook on evolutionary computation*. IOP Publishing Ltd. and Oxford University Press.

Boonlong, K., Chaiyaratana, N., & Kuntanapreeda, S. (2002a). Time optimal and time–energy optimal control of a satellite attitude using genetic algorithms. In

The 2002 ASME international mechanical engineering congress and exposition, New Orleans, LA, DSC-3343611-8.

Boonlong, K., Chaiyaratana, N., & Kuntanapreeda, S. (2002b). Using a co-operative co-evolutionary genetic algorithm to solve optimal control problems in a hysteresis system. In *The 2002 world congress on computational intelligence – The 2002, congress on evolutionary computation, Honolulu, HI* (Vol. 2, pp. 1504–1509).

Chang, P. C., Chen, S. H., & Fan, C. Y. (2008). Mining gene structures to inject artificial chromosomes for genetic algorithm in single machine scheduling problems. *Applied Soft Computing Journal, 8*(1), 767–777.

Chang, P. C., Chen, S. H., & Liu, C. H. (2007). Sub-population genetic algorithm with mining gene structures for flow shop scheduling problems. *Expert Systems with Applications, 33*(3), 762–771.

Eshelman, L. I. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. J. E. Rawlins (Ed.). *Foundations of genetic algorithm* (Vol. l, pp. 265–283). San Mateo, CA: Morgan Kanfmann.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In *Proceedings of the fifth international conference on genetic algorithms*.

Fonseca, C. M., & Fleming, P. J. (1995). Multi-objective genetic algorithms made easy: Selection, sharing and mating restriction. In *Proceedings of the first international conference on generic algorithms in engineering systems: Innovations and applications, Sheffield, UK* (pp. 45–52).

Fonseca, C. M., & Fleming, P. J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorithms – Pan I: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics, 28*(1), 26–37.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. New York: Wiley.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Massachusetts: Addison-Wesley, Reading.

Goldberg, D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications (ICGA'87)* (pp. 41–49).

Larrañaga, P., Kuijpers1, C. M. H., Murga1, R. H., Inza1, I., & Dizdarevic1, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review, 13*.

Mauldin, M. L. (1984). Maintaining diversity in genetic search. In *Proceedings of the national conference on artificial intelligence, Austin, TX* (pp. 247–250).

Mori, N., Yoshida, J., Tamaki, H., Kita, H., & Nishikawa, Y. (1995). A thermodynamical selection rule for the genetic algorithm. In *Proceedings of the second IEEE international conference on evolutionary computation, Perth, WA* (pp. 188–192).

Peña, J., Robles, V., Larrañaga, P., Herves, V., Rosales, F., & Perez, M. (2004). GA-EDA: Hybrid evolutionary algorithm using genetic and estimation of distribution algorithms. In *Innovations in applied artificial intelligence: 17th international conference on industrial and engineering applications of artificial intelligence and expert systems, IEA/AIE 2004, Ottawa, Canada, May 17–20*.

Pimpawat, C., & Chaiyaratana, N. (2001). Using a cooperative co-evolutionary genetic algorithm to solve a three-dimensional container loading problem. In *The 2001 congress on evolutionary computation, Seoul, Korea* (Vol. 2, pp. 1197–1204).

Potter, M. A., & DeJong, K. A. (1994). A cooperative co-evolutionary approach to function optimization. *Lecture Notes in Computer Science, 866*, 249–251.

Shimodaira, H. (1996). A new genetic algorithm using large mutation rates and population-elitist selection (GALME). In *Proceedings of the eighth IEEE international conference on tools with artificial intelligence, Toulouse, France* (pp. 25–32).

Shimodaira, H. (1997). DCGA: A diversity control oriented genetic algorithm. In *Proceedings of the second international conference on genetic algorithms in engineering systems: Innovations and applications, Glasgow, UK* (pp. 444–449).

Shimodaira, H. (1999). 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).

Shimodaira, H. (2001). A diversity-control-oriented genetic algorithm (DCGA): Performance in function optimization. In *The 2001 congress on evolutionary computation, Seoul, Korea* (pp. 44–51).

Tsutsui, S., Fujimoto, Y., & Ghosh, A. (1997). Forking genetic algorithms: GAs with search space division schemes. *Evolutionary Computation, 5*, 61–80.

Ursem, R. K., Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., & Zalzala, A. (1999). Multinational evolutionary algorithms. In *Proceedings of the congress of evolutionary computation (CEC-99)* (Vol. 3, pp. 1633–1640).

Weerayuth, N., & Chaiyaratana, N. (2002). Closed-loop time-optimal path planning using a multi-objective diversity control oriented genetic algorithm. In *The 2002 IEEE international conference on systems, man, and cybernetics, Hammamet, Tunisia* (Vol. 6, pp. 345–351).

Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproduction trials is best. In *Proceedings of the third international conference on genetic algorithms, Fairfax, VA* (pp. 116–121).