

## Research Article

# Hole Machining Path Planning Optimization Based on Dynamic Tabu Artificial Bee Colony Algorithm

<sup>1</sup>Wen-Qi Zhong, <sup>2</sup>Yuan-Biao Zhang, <sup>3</sup>Xiao-Yun Li, <sup>1</sup>Chun-Sheng Zheng and <sup>2</sup>Yan-Lan Zhang

<sup>1</sup>Electrical and Information Department, Jinan University, 519070, China

<sup>2</sup>Mathematical Modeling Innovative Practice Base, Packaging Engineering Institute, Key Laboratory of Product Packaging And Logistics of Guangdong Higher Education Institutes, Jinan University, Zhuhai, 519070, China

<sup>3</sup>Department of Child Health, University College, London

**Abstract:** By setting up the 'point milling process-TSPPC' model, this study firstly is to further discreteness the point milling process and then successfully transforms it into a traveling salesman problem with precedence constraints through setting the tabu list to record manufacturing procedures. Based on characteristics of the constraints, the tabu list is used to further expand the Bee Colony algorithm and get the Dynamic Tabu Artificial Bee Colony algorithm (DTABC-algorithm) which can dynamically update the solution space. The DTABC-algorithm is used to solve the 'point milling process-TSPWTR' model. By testing, not only the 'point milling process-TSPPC' model can effectively simplify the problem of point milling process, but also the DTABC-algorithm can show outstanding properties through the solving process.

**Keywords:** Dynamic tabu artificial bee colony algorithm, hole machining path planning optimization, the traveling salesman problem with precedence constraints

## INTRODUCTION

Point milling is a important components of printed substrate and its processing cost usually accounts for 30-40% of the total cost of plate making. The drilling machine is mainly used in point milling process during procedures of plate making. The point milling process by the drilling machine is part of classic Hole Machining Path Planning (HMPP) process: the production efficiency of HMPP is determined by the machining time and the auxiliary time of the drill moving from one whole site to another. In the Merchant's survey, it showed that the time of tools shifting and drill moving accounted for 70% of the average time for the whole manufacturing process. Therefore, HMPP optimization is the key to improving the production efficiency of the drilling machine. Presently, the drilling machine in practice is normally single drill bit process and different types of whole processing require different combinations of tools for point milling. Meanwhile, the production process results in the processing order of tools in some tool combinations. So the point milling process is a space-time composite dimension problem.

This problem is essentially belongs to the discrete optimization problem. Traditional solutions such as Brute-force method are hardly used in practical applications, for the reason that the combinatorial

explosion exists in the solution space while the number of points is unlimited. But modern artificial intelligence methods such as artificial neural network and genetic algorithms bring about new hope for the solution of this problem. Seely was the first one to propose the colony self-organization model. In 2005, based on living habits of bees, Karaboga applied the bee colony algorithm to solve problem of numerical optimization for the first time (Karaboga, 2005). In 2006, Teodorovic clearly put forward the bee colony algorithm (Teodorovic *et al.*, 2006). In 2007, B.Akay proposed the artificial bee colony algorithm (Karaboga and Akay, 2007). In 2008, Wong applied the bee colony algorithm to solve the traveling salesman problem and the study showed that the bee colony algorithm could get better results than other algorithms (Wong *et al.*, 2008). In Karaboga *et al.* (2011) further optimized the artificial bee colony algorithm and proposed the combinatorial artificial bee colony algorithm to solve the traveling salesman problem (Karaboga *et al.*, 2011). Researchers found that ABC algorithm had better advantages than PSO algorithm and ASO algorithm, such as faster convergence rates in global searching space and better global searching abilities in localities.

In this study we further disperse the problem of point milling and then transform it to the Traveling Salesman Problem with Precedence Constraints

(TSPPC) by setting up the 'point milling process-TSPPC' model. We propose an efficient ABC approach to solve the TSPPC which is formulated by the 'point milling process-TSPPC' model. Even though abundant contributions to the TPS have been done in the ABC research area, we cannot find any results for the TSPPC. The traditional ABC approaches are not suitable to solve the TSPPC because the precedence constraints cannot be handled adequately. In order to efficiently handle the precedence constraints, a new improved method is developed in our study. The key concept of the new improved method is to use a Tabu Sort (TS), which is defined as an ordering of vertices in a directed graph. Therefore, based on the studies conducted by Wong *et al.* (2008) and Karaboga *et al.* (2011) we further expand the bee colony algorithm to get the Dynamic Tabu Artificial Bee Colony (DTABC) algorithm and then use it to solve the problem of 'point milling process-TSPWTR'.

### THE 'POINT MILLING PROCESS-TSPPC' MODEL

In this section, we set up the 'point milling process-TSPPC' model to further disperse the point milling process and then simplify the initial problem to the problem of 'point milling process-TSPPC'.

The 'point milling process-TSPPC' model can be divided into two parts: discretization process and setting the Tabu list. The first step is the discretization process which matches the guided information with the machine operations as much as possible in order to decrease unnecessary expenses. Although computational difficulties exist during the discretization process when the number of element  $v$  in the operational set is more than that of the set including all points, the loss due to computational difficulties can be brought down dramatically if we use the Swarm intelligence algorithm. The second part is to record the timing sequence information of job information of point milling process by setting the tabu list. Point milling process is a space-time composite dimension problem. Precedence constraints in the problem of 'TSPPC' exist in terms of restraining the order of some points in time series. Based on such conditions, recording the timing sequence information of job information of point milling process by setting the tabu list is not only quick, but also efficient.

**Discretization process:** The minimal operation (the operation not needed to be further divided) in the point milling process includes drill moving, point milling by some kind of tool and tools changing. To complete the point processing of some holes pattern may include one

or more operations which are orderly or disordered. That means the point processing information is the summary of one or more pieces of operational information. Based on such theory, we can divide the point processing information of point into one or more pieces of operational information. The segmentation scheme is determined by the production process.

In order to better understand the discretization process, we define the set of all points as  $V$ . Assuming  $\exists \theta \in V$  while the point processing information  $\theta$  is  $(x, y, \varepsilon)$  in which  $x, y$  are the coordinate values of point  $\theta$  in the circuit board PCB and  $\varepsilon$  is the type of point needed to be processed by tools  $a, b, c$  ( $\varepsilon = \{a, b, c\}$ ), therefore, we get the set of operational information  $\lambda_n^\theta$ :

$$\lambda_n^\theta = \{x, y, a), (x, y, b), (x, y, c)\}, n = 1, 2, 3$$

$x, y$  are coordinate values of point  $\theta$  in the circuit board;  $n$  is determined by practical processing technology. When  $n$  equals to 1,  $\lambda_1^\theta = (x, y, a)$ .

By the discretization process, we can get the gathering set of sets of operational information which is defined as operation set  $v$ :

$$v = \{\lambda_n^\theta | \lambda_n^\theta = (x, y, o), \forall \theta \in V, \forall o \in \varepsilon\}$$

**Setting the tabu list:** Since the element  $\lambda_n^\theta$  in the operation set  $v$  is a set in itself, elements in the operation set  $\lambda_n^\theta$  are disordered. While  $\exists \theta \in V$  and  $\varepsilon$  in the processing information  $(x, y, \varepsilon)$  of point  $\theta$  is needed to be processed orderly processed by tools  $a, b, c$ , we set the tabu list to record the operational order of point milling process to avoid missing the timing sequence information of processing information of points. Hereby, we define that  $tabu_{(i,j)}$  records the  $j^{\text{th}}$  operating information  $\lambda_j^\theta$  of the  $i^{\text{th}}$  order point  $\theta$ , then  $tabu_{(i,j+1)}$  records the  $(j+1)^{\text{th}}$  operating information  $\lambda_{j+1}^\theta$  of the  $i^{\text{th}}$  order point  $\theta$ . Besides, according to the processing requirement of the order point  $\theta$ , the processing of  $\lambda_j^\theta$  should be ahead of  $\lambda_{j+1}^\theta$ .

Therefore, the timing sequence information of point milling process is recorded in the tabu list matrix. In order to improve the solving performance of our model, we choose to optimize the data structure of the tabu list. Not all the elements in the tabu list matrix but elements not in the first row are needed to be tabooed. The first row acts as the index to show the operation can be carried out. When the element in the first row of each line is carried out, the element will be deleted and the following element in the same line will be moved forward. For example,  $tabu_{(i,:)}$  stands for line  $i$  in the tabu list matrix corresponding with the number  $i$  of

order point  $\theta$ . Assuming  $tabu_{(i,:)} = [\lambda_1^\theta \lambda_2^\theta \lambda_3^\theta]$ , the set of optional points is  $A_t \cup \{\lambda_1^\theta\}$ . After  $\lambda_1^\theta$  is carried  $tabu_{(i,:)} = [\lambda_2^\theta \lambda_3^\theta]$  out, and the set of optional points will be  $A_t \cup \{\lambda_2^\theta\}$ .

**'Point milling process-TSPPC' model:** In the model, each piece of operational information is equivalent to each point in the problem of TSPPC. Meanwhile, expenses by operations are generated by drill moving and tools changing. Therefore, we define  $Cost^{all} = Cost^{distance} + Cost^{change}$ , in which  $Cost^{distance}$  means cost due to drill moving and  $Cost^{change}$  means cost due to tool changing. Because of the tabu list, we introduce a matrix variable  $y_{(ij)}$  and define it as:

$$y_{(ij)} = \begin{cases} 1 & \text{otherwise} \\ 0 & \text{if vertex } j \text{ can not be visited immediately after vertex } i \text{ from Tabu list} \end{cases}$$

To conclude, we get the 'point milling process-TSPPC' model as follows:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in \phi} Cost^{all} x_{(i,j)} y_{(i,j)} \quad ; \\ \text{s.t.} \quad & \sum_{j \in v} x_{(i,j)} y_{(i,j)} = 1 \quad , i \in v \\ & \sum_{j \in v} x_{(j,i)} y_{(j,i)} = 1 \quad , i \in v \end{aligned}$$

$v$  : The set of operations

$X_{(ij)}$ : The connection matrix, which indicates operation

$i$  : Connected to operation  $j$  and operation  $i$  is prior to operation  $j$

$y_{(ij)}$  : The tabu list matrix variables

$\phi$  : The choice of process path

### DYNAMIC TABU ARTIFICIAL BEE COLONY ALGORITHM (DTABC-ALGORITHM)

**Artificial Bee Colony (ABC) algorithm:** The Artificial Bee Colony (ABC) algorithm is a heuristic swarm intelligence algorithm based on foraging behaviors of bees. In the ABC algorithm, the bee colony acts as three types of roles: employed bees, onlooker bees and scout bees. Each employed bee is used to point out the corresponding food source which is the optimal path. Employed bees will save the information of food source and go back to the honeycomb and then recruit onlooker bees by the waggle dance. Through observing the waggle dance, onlooker bees will decide which food source to explore and try to improve the quality of food source while exploring. If employed bees give up the food source, they will transform into scout bees and

search for new food sources randomly. Basic steps of the ABC algorithm can be described based on the following pseudo algorithm:

- Initialize the parameters: Colony Size (CS), Maximum Number of Iterations (MI).
- Initialize the food source:  $\phi_i$  where  $i = 1, 2, \dots, CS$
- For  $i = 1$  to MI
  - Evaluate the population of solutions
  - Memorize the best solution
  - Employed-bees-lead-or-abandon ()
  - Waggle-dance ()
  - Onlooker-bees-search ()
  - Scout-bees-search ()
- End-for

Based on abundant studies in solutions to the problem of TSP by the ABC algorithm and also analysis of properties of such solutions conducted by (Wong *et al.*, 2008; Karaboga *et al.*, 2011) findings show that ABC algorithm has better advantages than PSO algorithm and ASO algorithm, such as faster convergence rates in global searching space and better global searching abilities in localities. Meanwhile, there is little difference between the optimal solution by the ABC algorithm and the given standard. So we can conclude that ABC algorithm is not only suitable but efficient for solving the problem of TSP.

### Dynamic Tabu Artificial Bee Colony (DTABC) algorithm:

However, the problem of 'point milling process-TSPPC' is not a traditional problem of TSP. Because of the precedence conditions, the initial problem is changed into a space-time composite dimension problem. Traditional ABC algorithm cannot solve the problem of TSPPC. So we propose the Dynamic Tabu Artificial Bee Colony (DTABC) algorithm which improved on the basis of the traditional ABC algorithm. The DTABC algorithm is to solve the space-time composite dimension problem of point milling process by setting the tabu list to depict time dimension and also dynamically update the solution space.

In order to explain clearly, we illustrate the process of searching for food sources by observing onlooker bees. Assuming onlooker bees  $k$  is searching for the food source, the path of leader bee will be  $\phi_i^{Leader} = \{3\ 2\ 1\ 4\ 3\}$  and the path of onlooker bees  $k$  will be  $\phi_i^{Onlooker} = \{3\}$  and the set of points not searched is  $A_k, A_k = \{1\ 2\ 4\}$ . Meanwhile, the tabu list matrix is  $Tabu_k = [2\ 4]$ . So the probability of next searching point is  $P_k^{Onlooker} = \{\alpha\ \beta\ 0\}$ , in which the values of  $\alpha, \beta$  are determined by different searching rules. If the chosen point is  $\{2\}$ , then  $\phi_i^{Onlooker} = \{3\ 2\}$ ,  $A_k = \{1\ 4\}$ ,  $Tabu_k = [4]$ . When onlooker bees  $k$  chooses next

Table 1: Lookup table for adjustment of  $Pf_{onlooker}$

Profitability scores	$Pf_{onlooker}$
$0.85Pf_{colony} \leq Pf_i$	0.60
$0.85Pf_{colony} \leq Pf_i < 0.85Pf_{colony}$	0.30
$0.5Pf_{colony} \leq Pf_i < 0.65Pf_{colony}$	0.20
$Pf_i < 0.5Pf_{colony}$	0.00

point again, the probability of next searching point will be  $P_k^{onlooker} = \{a, \chi\}$ , in which values  $a, \chi$  of will be determined by different searching rules. The pseudo algorithm of searching for food sources by onlooker bees using the DTABC algorithm can be described as follows:

```

Onlooker-bees-DTsearch ()
{
  Initializing-ResultZoom ()
  Onlooker-bees-NewSearch ()
  Renew-Tabu
}
    
```

In practice, the proportion of three types of bees has two conditions-constant and dynamic variations, which is determined by actual situations. According to DTABC algorithm in this study, while deciding the proportion of three types of bees, we take the problem of dynamics into account and choose the method which can react correspondingly to the present results. Regarding the following probability of onlooker bees, have done relevant studies (Nakrani and Tovey, 2004; Chong *et al.*, 2006). Based on their findings, we cope with the following probability as follows (Table 1):  $Pf_i$  Represents the profitability score the  $i^{th}$  bee and is defined as formula (1):

$$Pf_i = \frac{1}{L_i}, \quad L_i = \text{tour length} \tag{1}$$

$Pf_{colony}$  Represents the average profitability score of the whole bee colony and is defined as formula (2):

$$Pf_{colony} = \frac{1}{n} \sum_{i=1}^n Pf_i = \frac{1}{n} \sum_{i=1}^n \frac{1}{L_i} \tag{2}$$

where,

$L_i$ : The cost of path  $\phi_i$  of the  $i^{th}$  bee

$n$ : The number of bees,  $n = CS$

The process of searching by scout bees is similar to the process of searching for food sources by onlooker bees, so it can be obtained by analogy:

```

Scout-bees-DTsearch ()
{
  Initializing-ResultZoom ();
    
```

```

  Scout-bees-NewSearch ();
  Renew-Tabu;
}
    
```

Based on the renewed idea mentioned above, the pseudo algorithm of the DTABC algorithm is as below:

- Initialize the parameters: Colony Size (CS), maximum number of iterations (MI).
- Initialize the food source:  $\phi_i$  where  $i = 1, 2, \dots, CS$
- For  $i = 1$  to MI
  - Evaluate the population of solutions
  - Memorize the best solution
  - Employed-bees-lead-or-abandon ()
  - Waggle-dance ()
  - Onlooker-bees-DTsearch ()
  - Scout-bees-DTsearch ()
- End-for

## EXPERIMENTS AND RESULTS

In this section, we first illustrate the calculation example and then simplify the problem by the 'point milling process-TSPPC' model and solve the problem by the DTABC algorithm. Finally, we verify the properties of the algorithm by comparing with the modified genetic algorithm.

**Running results:** In order to solve the problem of practical example, we choose 30 points including three points from each of ten types of points from the online datab (<http://www.mcm.edu.cn/uploadcn/node/187/Cngox PPI9089a 673701f3be 3eb90b9b 10a3b2dee.rar>) and further disperse and simplify the problem by using the 'point milling process-TSPPC' model. At last, we adopt the DTABC algorithm to solve the problem. The main parts of the model are interpreted in details in section two and three and only the results from the DTABC algorithm by adopting the 'point milling process-TSPPC' model are illustrated here in section four.

Parameter settings are shown below where LimitMax means the max number of optimal solution:

Parameter	Value
CS	200
MI	3000
LimitMax	100

### Running results:

Original Point Number	30
Point milling process-TSPPC'	54
Model Point Number	
Best Result (Unit: RMB)	62.933
Average optimal solution by repeating 50 times (unit: RMB)	63.055

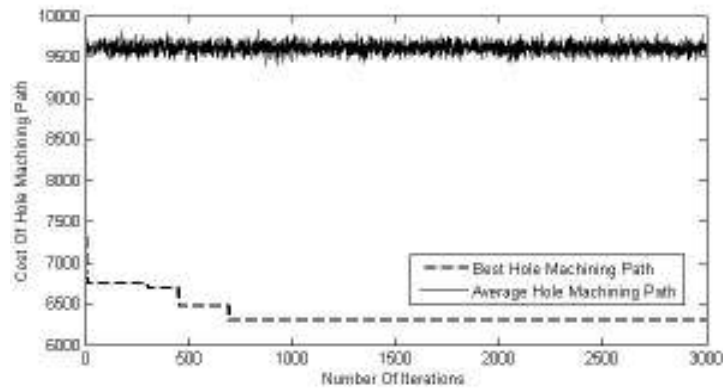


Fig. 1: Hole machining path

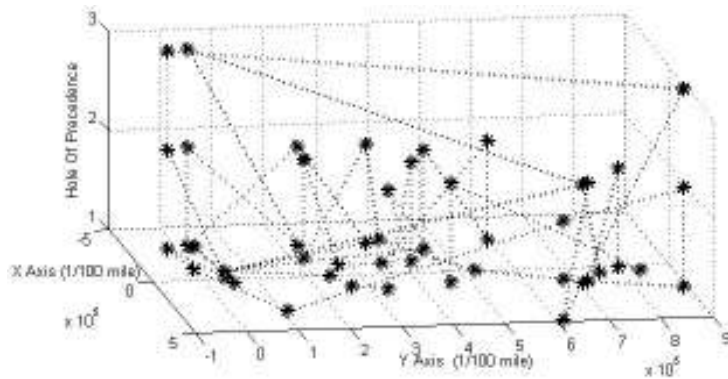


Fig. 2: Processing path

From the running results, we can see that the 'point milling process-TSPPC' model can disperse and simplify the problem (Fig. 1, 2). The simplified problem is solvable and can meet practical processing requirements. Meanwhile, the DTABC algorithm has its unique advantages in solving the simplified problem (problem of TSPPC): not only the rate of solving and convergence is fast, but also the error between the average optimal solution by repeating 50 times and the best result is small. The running results of the optimal solutions are stable, which means that applying the DTABC algorithm to solve the 'point milling process-TSPPC' model is practical and efficient.

**Performance comparison:** At the present stage, there is no accepted solution to the problem of TSPPC. But Scientists such as Moon (Chiung *et al.*, 2002) has put forward a new algorithm based on the genetic algorithm and obtained better results. Therefore, we decide to run a solving performance test of DTABC algorithm using Moon's example. In Moon's example, there are 20 points and 31 precedence constraints. Precedence constraints are shown in Fig. 3. The assumed distance between points is shown in Table 2.

In order to solve the problem and have a comparison with results done by Chiung *et al.* (2002), parameters are set as follows: CS = 100, MI = 500, LimitMax = 100. The results solved by the DTABC algorithm are as below:

6-1-2-7-5-11-8-14-4-3-18-12-10-16-19-15-17-13-9-20

The path length is 52.

In order to contrast the properties of two algorithms in solving the problem of TSPPC, comparisons of results are shown in the Table 3.

By comparing the Table 3 and the Fig. 4, we find that the DTABC algorithm is superior to the genetic algorithm proposed by Moon. In terms of the optimal solution, the result by DTABC algorithm is 52, which is 14.75% less than that of Moon's genetic algorithm. Regarding the convergence, the optimal solution by DTABC algorithm appears in the 65<sup>th</sup> generation, which is 370 generations earlier than that of Moon's genetic algorithm. By the analysis, we find that DTABC algorithm is superior to Moon's genetic algorithm not only in searching for the optimal solution, but also in terms of convergence. Therefore, the DTABC

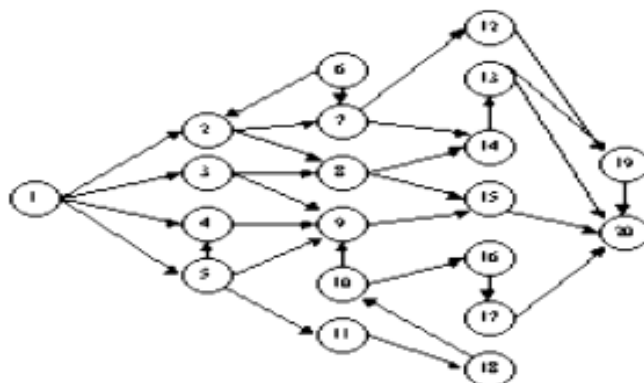


Fig. 3: TSP network with 20 vertices and 31 precedence constraints (Chiang *et al.*, 2002)

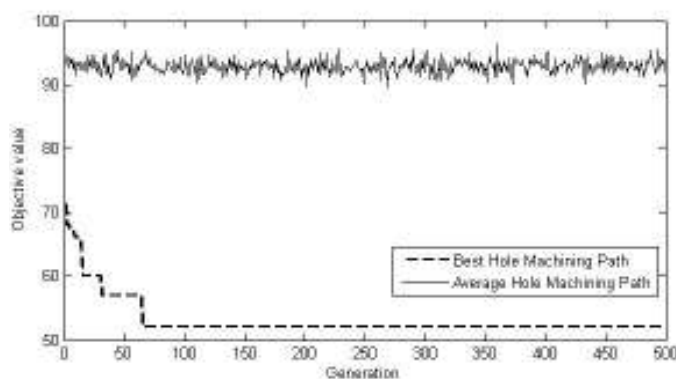


Fig. 4: Hole machining path

Table 2: Transition distance between vertices (Chiang *et al.*, 2002)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		2	1	4	1	2	11	9	1	3	7	12	5	8	8	8	13	7	2	12
2			12	12	4	6	2	8	6	7	14	1	11	9	2	6	13	14	1	3
3				5	8	2	5	9	7	8	13	6	9	2	6	6	14	2	4	5
4					6	6	11	12	5	11	4	5	11	1	3	1	17	1	14	14
5						7	2	13	3	1	6	7	14	8	7	7	5	1	8	13
6							2	6	14	6	9	3	7	13	13	3	13	1	13	3
7								11	14	6	4	1	7	6	12	9	1	8	5	4
8									14	14	3	11	1	1	3	1	6	5	14	14
9										3	12	12	2	12	4	2	14	13	11	7
10											8	4	8	5	4	4	3	6	12	11
11												5	9	4	9	9	6	12	14	11
12													7	6	1	13	7	1	6	8
13														1	1	4	1	12	4	6
14															4	9	12	4	9	13
15																1	1	1	1	5
16																	8	5	2	14
17																		14	7	1
18																			5	6
19																				14

Table 3: Comparisons of results

	DTABC algorithm	Proposed GA by moon
The optimal solution	6-1-2-7-5-11-8-14-4-3-18-12-10-16-19-15-17-13-9-20	6-1-2-7-5-11-4-3-18-12-10-9-16-17-8-14-13-19-15-20
Length of optimal solution	52	61
Convergence generations of optimal solution	65	430

algorithm applied in solving the problem of TSPPC has its own benefits.

### CONCLUSION

In this study, we set up the 'point milling process-TSPPC' model to simplify the problem of point milling

process and transform it into the problem of TSPPC. In order to better solve this problem, we expand the ABC algorithm and obtain the DTABC algorithm. With testing the optimal solution by multiple loops, we find the DTABC algorithm is suitable for solving the problem of TSPPC. Results of the optimal solution by the DTABC algorithm fluctuate slightly and the speed of solving is fast. So the DTABC algorithm has inimitable advantages. Finally, through comparing with the genetic algorithm in solving the problem of TSPPC, we find that the DTABC algorithm is superior to Moon's genetic algorithm not only in searching for the optimal solution, but also in terms of convergence. As a future work, we will consider conducting studies in the parameter setting and convergence of the DTABC algorithm so that it will have wider application prospects.

#### REFERENCES

- Chiung, M., J. Kim, G. Choi and Y. Seo, 2002. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *Eur. J. Oper. Res.*, 140: 606-617.
- Chong, C.S., Y.H. Malcolm, A.I. Low, Sivakumar and K.L. Gay, 2007. Using a bee colony Algorithm for neighborhood search in job shop scheduling problems. *Proceeding of 21st European Conference on Modeling and Simulation (ECMS 2007)*.
- Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Karaboga, D. and B. Akay, 2007. An artificial bee colony (abc) algorithm on training artificial neural networks. In *15th IEEE Signal Processing and Communications Applications, SIU 2007 E Skisehir, Turkiye*, (pp: 1-4).
- Karaboga, D., G. Beyza and T. Kayseri, 2011. A combinatorial artificial bee colony algorithm for traveling salesman problem (J). *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on (C)*, 2011: 50-53.
- Nakrani, S. and C. Tovey, 2004. On honey bees and dynamic server allocation in Internet hosting centers. *Adaptive Behavior*, 12(3-4): 223-240.
- Teodorovic, D., P., Lucic, G. Markovic and M.D.Orco, 2006. Bee Colony Optimization: Principles and Applications Neural Network Applications in Electrical Engineering (A), *NEUREL 2006. 8th Seminar on 2006 (C)*, pp: 151-156.
- Wong, L.P., Y.H.L. Malcolm and S.C. Chin, 2008. A bee colony optimization algorithm for traveling salesman problem (A). *Second Asia International Conference on Modelling and Simulation (C)*, 2008: 818-823.