

# Path optimization of CNC PCB drilling using hybrid Taguchi genetic algorithm

Path  
optimization  
of CNC  
PCB drilling

107

Dony Hidayat Al-Janan

*National Kaohsiung First University of Science and Technology,  
Kaohsiung, Taiwan, ROC, and*

Tung-Kuan Liu

*Institute of Engineering Science and Technology,  
National Kaohsiung First University of Science and Technology, Taiwan, ROC*

## Abstract

**Purpose** – In this study, the hybrid Taguchi genetic algorithm (HTGA) was used to optimize the computer numerical control-printed circuit boards drilling path. The optimization was performed by searching for the shortest route for the drilling path. The number of feasible solutions is exponentially related to the number of hole positions. The paper aims to discuss these issues.

**Design/methodology/approach** – Therefore, a traveling cutting tool problem (TCP), which is similar to the traveling salesman problem, was used to evaluate the drilling path; this evaluation is considered an NP-hard problem. In this paper, an improved genetic algorithm embedded in the Taguchi method and a neighbor search method are proposed for improving the solution quality. The classical TCP problems proposed by Lim *et al.* (2014) were used for validating the performance of the proposed algorithm.

**Findings** – Results showed that the proposed algorithm outperforms a previous study in robustness and convergence speed.

**Originality/value** – The HTGA has not been used for optimizing the drilling path. This study shows that the HTGA can be applied to complex problems.

**Keywords** Drilling path, Hybrid Taguchi genetic algorithm, Travelling cutting tool problem

**Paper type** Research paper

## 1. Introduction

In recent decades, printed circuit boards (PCBs) have become a crucial product because of a sharp increase in demand for electronic devices. However, the time consumed by computer numerical control (CNC) during PCB drilling poses a serious problem: it reduces the efficiency of the drilling process. To enhance the efficiency and obtain the minimum trace, the movement of the drilling tool on the CNC should be controlled. In a single case of two-dimensional CNC PCB drilling, all hole points have the same priority; to improve the efficiency of the drilling process, the hole points should be assigned priorities.

A simple method for searching for the shortest drilling path is to list all feasible routes for the path and then choose the minimum trace. This technique may take a long time because the number of solutions increases exponentially with the number of holes. For instance, five hole positions correspond to 120 (5!) feasible sequences, but for six hole positions, the number of feasible sequences increases sharply to 720 (6!) (Lim *et al.*, 2014).

This study was done during doctoral program in National Kaohsiung First University of Science and Technology, Taiwan, ROC, which is supported by Indonesian Government in scheme BPPLN DIKTI 3+1 that began on September 2013.



Several researchers have studied the problem of determining the minimal sequence of CNC drill movements by using stochastic methods. Nevertheless, stochastic methods do not guarantee the optimal solution; they only provide a solution close to the optimal solution. However, they continue to be used because they are not time consuming.

To determine the optimal drilling path, the traveling salesman problem (TSP) should be used to solve the CNC path problem, which is named the traveling cutting tool problem (TCP), because of the similarity between these two problems (Qudeiri *et al.*, 2013). In solving the TSP, the final path must return to the initial city. In the TCP, the path need not return to the first point; the only objective is to determine the shortest path. For instance, assuming that the shortest path for drilling five points is [4 5 2 3 1], for the subsequent processes, the paths are [1 4 5 2 3], [3 1 4 5 2], and so on.

Many methods of TCP optimization have been studied by several researchers. Zhu and Zhang (2008) employed the particle swarm method to minimize the CNC drilling path. Furthermore, Lim *et al.* (2012, 2014) used cuckoo search (CS), which was inspired by the egg selection method of the cuckoo, for searching for the shortest CNC drilling path. Genetic algorithms (GAs), which are classical searching methods, have been developed for obtaining the optimal path (Al-Sahib and Abdulrazzaq, 2014; Mansour *et al.*, 2013; Qudeiri *et al.*, 2006, 2013).

Many types of techniques, including a hybrid GA (Gupta *et al.*, 2011), parallel ant colony (Montiel-Ross *et al.*, 2012), and hybrid ant colony together with a GA (Abbas *et al.*, 2014), have been used in previous studies. Research on path optimization has been inspired by not only biological phenomena but also physical phenomena. Omar *et al.* (2014) proposed the gravitational search algorithm for solving the path optimization problem. The Euclidean TSP (Qudeiri *et al.*, 2013) and simulated annealing hybrid GA (Ye and Rui, 2013) have also been used for optimizing the path of a drilling tool. Additionally, Route Optimizer 3, which is based on an evolutionary algorithm, was used by Sigl and Mayer (2005) to optimize the drill route.

In this study, three problem cases were considered to analyze the efficiency of the hybrid Taguchi genetic algorithm (HTGA) in obtaining an optimal PCB drilling path. These three problems are worth studying because they are based on real problems related to PCB drilling and many previous researchers have used them to verify the efficiency of path optimization methods. Lim *et al.* (2012) solved Workpiece 1 (with nine hole positions) by using CS as the optimizer; Zhu and Zhang (2008) had previously studied the same problem case by using particle swarm optimization (PSO) as the solver. Workpiece 2 has been solved using gravitational search (Omar *et al.*, 2014), CS (Lim *et al.*, 2012), the firefly algorithm (Ismail *et al.*, 2012), and PSO (Zhu and Zhang, 2008). Workpiece 3 was solved using a GA (Wei *et al.*, 2008). Lim *et al.* (2014) considered these three problems for optimizing the PCB drilling path by using CS. Therefore, in the current study, the performance of the HTGA (Tsai *et al.*, 2004) in optimizing a two-dimensional PCB drilling path was compared with that of CS (Lim *et al.*, 2014).

The CS algorithm, developed by Yang and Deb (2009), was inspired by the unique proliferation pattern of the cuckoo bird. A cuckoo lays its egg in other birds' nests for breeding. CS requires three idealizations: first, each cuckoo lays an egg in a randomly selected nest; second, the superior cuckoo egg, which has superior chromosomes compared with those of the host birds' eggs, hatches and the young cuckoo becomes part of the next generation; third, the cuckoo's egg may fail to hatch if detected by the host bird; the detection probability is denoted by  $p_a$  ( $\in [0,1]$ ). When  $p_a = 0$ , means host bird never found

cuckoo egg; while if  $p_a = 1$ , means host bird always found cuckoo egg. The cuckoo egg and host birds' eggs compete to be the best egg. The egg with the best chromosomes for achieving the highest fitness wins this competition. To create eggs superior to the host birds' eggs, the chromosomes of cuckoo egg are produced using Levy's flight (Yang and Deb, 2009).

Levy's flight was inspired by the jumping of animals and insects from one point to another. The distance of a jump is ascertained by determining a previous point in accordance with the Markov chain law (Yang, 2010). Levy's flight is affected by the host birds' eggs, which serve as the baseline (Yang and Deb, 2009). The number of cuckoo eggs is equal to the number of nests. Hence, the performance of CS is affected by the number of nests and  $p_a$ . The number of population members in the next generation produced in each iteration is twice that in the present generation, which ranks lower than the next generation.

According to the preceding discussion, new superior chromosomes (eggs) quickly adapt to the selection process and probability ( $p_a$ ) to change. If a new chromosome is superior to the existing chromosome and has a probability ( $p_a$ ), it is replaced immediately. This procedure may be risky if the chromosome has high diversity, possibly being trapped in local optima because of a concise selection process.

The HTGA uses a different strategy to produce the best chromosome. This method is similar to the traditional GA. The algorithm starts by creating a parent chromosome. This step is based on the number of population members, and it is similar to a step in CS. Subsequently, several chromosomes of the population are produced in crossover and mutation operations, and finally, in the elimination step, chromosomes are selected according to their fitness rank. In the HTGA, in addition to the original GA, two levels of the Taguchi orthogonal array (OA) are introduced between crossover and mutation for minimizing the error value and limiting the effect of unsuccessful target search during the crossover operation as an exploitation function. Only two chromosomes from the offspring population are analyzed. The number of generated chromosome candidates depends on the number of genes in a chromosome. Thus, more complex genes will be evaluated in more experiments. As a complement, a chromosome is mutated by using a neighbor method for preventing the algorithm from being trapped in local optima. Therefore, a complex chromosome or a function with many local optima produces a highly feasible superior chromosome (Tsai *et al.*, 2004).

This paper is organized as follows: it begins with a background of the study and a rough comparison between CS and the HTGA. Next, the problem of this study is defined and the application of the HTGA to it is described. The procedure of the HTGA is explained in the subsequent section. A discussion and conclusion regarding the experimental procedure are then presented. Finally, a comprehensive reference list is provided.

## 2. Problem definition

In real applications, the work table used for CNC drilling can move in the  $x$  and  $y$  directions. The movement in each direction is controlled by a stepper motor, and the drilling tool position is fixed. The drilling tool moves only vertically during the drilling process. The table is appropriately moved to set the hole positions underneath the drilling tool during the drilling process. The table movement cannot be calculated precisely because of the motor switching among acceleration, full speed, and deceleration; the table movement is therefore time dependent.

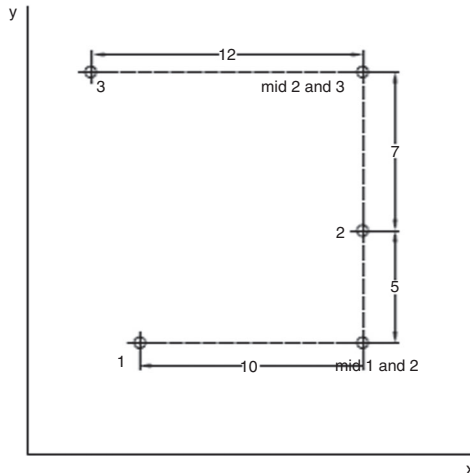
Lim *et al.* (2014) used the CS algorithm to optimize the CNC PCB drilling hole position sequence for three workpieces (WPs). The velocities in the  $x$  and  $y$  directions, which are controlled by two stepper motors, are assumed constant. The assumptions in this study were as follows:

- (1) the rotation speed of the stepper motors is constant;
- (2) all holes have the same diameter (no drilling change);
- (3) the distances from the initial point to the first hole position and from the final hole position to the initial point are neglected;
- (4) there is no time delay between the stepper motor and drilling movement; and
- (5) the drilling tool passes through each hole position only once.

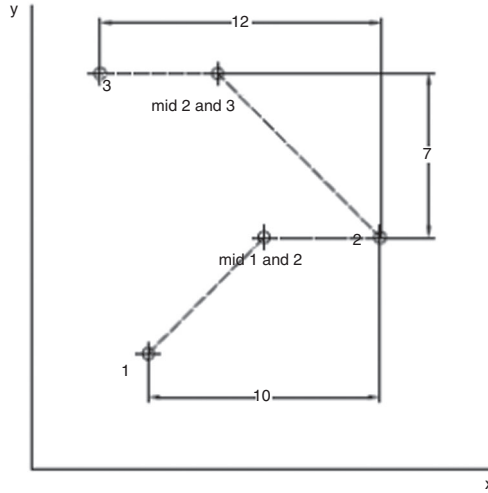
In this study, two cases of stepper motor movement were considered. In Case 1, two work table stepper motors operate separately. The movement is initially in the  $x$  direction and then in the  $y$  direction. Figure 1 shows the drill moving from 1 to 2; the drill moves from Point 1 to the midpoint between Points 1 and 2, finally halting at Point 2. In Case 2, two stepper motors operate simultaneously if the coordinates of the initial and destination points are such that movements along both axes are required. Figure 2 shows the drill shifting from Point 1 to 2; it moves from Point 1 to the midpoint between Points 1 and 2 because of the combined operation of the two stepper motors. Finally, the movement of the drill from the midpoint between Points 1 and 2 to Point 2 results from the operation of the stepper motor in the  $x$  direction (Lim *et al.*, 2014).

The time taken for all work table movement in the two cases was determined as follows:

$$\text{Case 1: } t = \sum_{i=1}^{n-1} \left( \frac{|x_i - x_{i+1}|}{v_x} + \frac{|y_i - y_{i+1}|}{v_y} \right) \quad (1)$$



**Figure 1.**  
Work table  
movement for case 1



**Figure 2.**  
Work table  
movement for case 2

$$\text{Case 2: } t = \sum_{i=1}^{n-1} \max \left( \frac{|x_i - x_{i+1}|}{v_x}, \frac{|y_i - y_{i+1}|}{v_y} \right) \quad (2)$$

These equations describe the total time of a route job. The parameter  $t$  is the total time taken for covering the entire route for  $n$  hole positions (second),  $i$  is the number sequence of a hole,  $x_i$  and  $x_{i+1}$  are the  $x$  coordinates of the initial and destination points in millimeters,  $y_i$  and  $y_{i+1}$  are the  $y$  coordinates of the initial and destination points in millimeters, and  $v_x$  and  $v_y$  are the drill velocities in the  $x$  and  $y$  directions (mm/s). Because  $v_x$  and  $v_y$  are set to be 1 mm/s, the distance and the time taken (TR) have the same value, but different units (Lim *et al.*, 2014).

### 3. Use of HTGA for TCP optimization

The HTGA was shown to be an effective method for searching for real optimal or optimal solutions with high robustness and quick convergence (Tsai *et al.*, 2004). In this study, the HTGA was applied to optimize the CNC drilling path for two-dimensional PCB drilling. The following paragraphs explain Taguchi's experimental design and the HTGA stepping procedure.

The Taguchi design of experiment (DOE) was along with an OA to reduce the number of experiments. The OA has a power ability for determining the optimal setting of input parameters on the base of a number of experiments less than the full factorial. The signal-to-noise ratio (SNR) formulation in the Taguchi OA is used to verify the robustness of the parameter setting in the array and to recommend the optimal setting if the setting in the array is not sufficiently optimal (Tsai *et al.*, 2004).

For instance,  $3^4$  (81) experiments are required to analyze three levels in four input parameters by using the full factorial. However, if the variable problem is evaluated using an OA, only nine experiments ( $L_9$ ) are required. This is because of fractional DOE, which is used in the OA method (Fowlkes and Creveling, 1995).

In this study, two levels of an OA were employed. A hole position was described as a gene, while a route or path was denoted as a chromosome. Many OAs were selected

using many genes, with the number of genes being equal or greater than the number of OAs. For instance, if it is assumed that a WP has seven points, then eight OA experiments ( $L_8$ ) are chosen. Generally, Taguchi's OA procedure can be formulated as  $L_n = 2^{(n-1)}$ , implying that for  $n$  experiments,  $(n-1)$  parameters in two levels can be analyzed. Table I shows an  $L_8$  OA. The column run provides the number of experiments. The numbers 1 and 2 in each column denote the parameter level, and the letters A to G denote the parameter set in each experiment.

The SNR is presented as a complement of the Taguchi DOE. The SNR is used to determine a robust parameter setting. There are three criteria for the SNR: the smaller the better, nominal is the best, and the larger the better. This study used the smaller the better for determining the optimal TR for CNC PCB drilling. This criterion involves the following formula:

$$\eta = -10\log \left( \frac{1}{n} \sum_{t=1}^n \frac{2}{y_t} \right) \tag{3}$$

where  $\eta$  is the SNR,  $n$  is the number of samples, and  $y_t$  denotes the data (Fowlkes and Creveling, 1995).

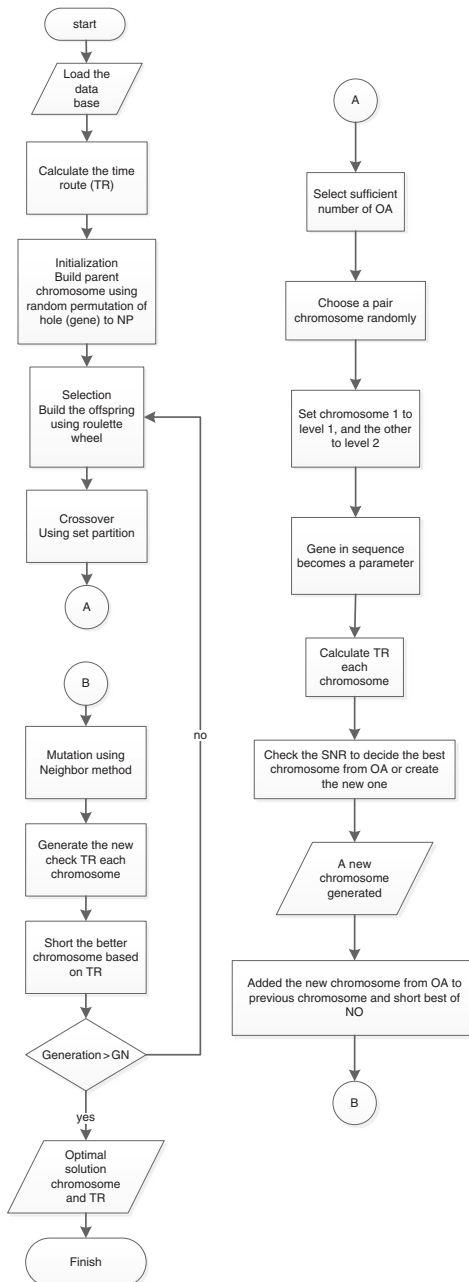
This procedure was used along with the SNR criteria to determine a robust setting; a more robust setting has a higher SNR. In Taguchi's OA, the number of chromosomes is equal to the number of experiments. Whole chromosomes are compared to determine the high SNR. Sometimes, the best chromosome is not present in the OA because of the OA not being the full factorial of the DOE. If this situation occurs, then a new chromosome will be proposed based on the higher of SNR.

Figure 3 describes the procedure for optimizing the TCP by using the HTGA:

- Step 0 Parameter setting. Input: data on hole position coordinates, number of population members (NP), number of offspring (NO), crossover rate (CR), mutation rate (MR), and number of generations (GN). Output: the best chromosome is the path route sequence having the smallest fitness value.
- Step 1 Determining the total time route (TR) for each pair of positions by using Equation (1) or (2), depending on the chosen case.
- Step 2 Initializing or generating the parental chromosome. In this step, each parental chromosome is produced using a random gene permutation. Each chromosome has a unique gene sequence (drilling point route), and therefore, the chromosomes are distinct from each other. The chromosome production process continues until the NP of the chromosomes is reached.

Run	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

**Table I.**  
 $L_8$  orthogonal array



**Figure 3.**  
Flowchart of  
the HTGA for  
optimizing the TCP

- 
- Step 3 Selection process involving a roulette wheel. In this process, the cumulative probability contribution of each chromosome is evaluated. Summation accumulation is described by a roulette wheel that is rotated to select the chromosome. This selection step is terminated when the number of chromosomes equals the NO. Chromosomes are selected in pairs and should be distinct from each other.
- Step 4 Crossover process using a set partition method. This step is applied to each chromosome pair chosen in Step 3, but it depends on the random number. If the random number is lower than the CR, this pair is crossed.
- Step 5 Taguchi DOE (explained in the next section).
- Step 6 Mutation by conducting neighbor search. Every chromosome obtained from Step 5 is mutated if the random number is lower than the MR.
- Step 7 A new superior chromosome is produced.
- Step 8 Eliminating the best NP chromosomes for maintaining high fitness. Steps 3-8 are repeated until the GN is reached.
- Step 9 Displaying the best chromosome and fitness value.

These steps are explained as follows:

1. Initialization: three data problem cases containing data on hole coordinates in two dimensions are considered. Each datum is used as an input for the HTGA. A chromosome represents a drilling route, whereas a gene denotes a hole coordinate. First, the TR should be calculated. The TR value depends on the case used (Case 1 or 2). The chosen case affects the time taken for moving the drill from one position to another. The TR calculation result is a square matrix ( $NP \times NP$ ) and is used to determine the TR for each position.

Next, a parental chromosome is produced using a random permutation of hole position numbers. The number of parent chromosomes is the NP, and hole positions are represented by genes. A higher NP renders the solution more precise, but requires a longer processing time. The TR of each chromosome is determined to create a roulette wheel representing the probabilities of chromosomes contributing to the overall probability.

2. Selection: the TR of each chromosome produced through the preceding process is used for creating the roulette wheel, which describes the probability of each chromosome that contributes to the overall probability. The highest value of the cumulative sum of probabilities is 1 (100 percent).

Selection refers to the process of selecting several chromosomes until the number of chromosomes selected equals the NO. In this study, the NO was determined as  $0.8 \times NP$ . Pairs of chromosomes were formed from the selected chromosomes by using a random number, which was used for determining the chosen chromosome from the position of the roulette wheel. The random number generated value (0-1), such that correlated with position in roulette wheel. The pairing of identical chromosomes must be avoided. Figure 4 shows the roulette wheel that determines the selected chromosome. In this figure, the NP is assumed to be eight chromosomes, and each chromosome contributes to the overall probability.

3. Crossover: the partition method was used for crossover. This method begins with a random permutation of four genes in each chromosome pair. Figure 5 shows the parental chromosomes P1 and P2 having ten genes. U1 and U2 are created by choosing four genes randomly from each chromosome. A gene (U1) is randomly chosen from P1. This randomly chosen gene is used along with the sequence in P2 to create a child chromosome (C1). Finally, a gene is randomly chosen from P2 and used together with the sequence in P1 to create C2.



4. Taguchi DOE: the Taguchi OA is used between crossover and mutation to generate superior chromosomes. This step begins with randomly choosing two chromosomes from the chromosomes produced in the crossover operation; and the two chromosomes are then considered to represent two levels of experimental parameters. Otherwise, the number of chromosomes (hole position sequence) becomes the input parameter of the DOE. The complete procedure of the Taguchi DOE algorithm is as follows:

Step 1 An adequate OA is selected on the basis of the number of genes (hole positions). The two levels of the OA are correlated, and the number of experiments should be equal to a power of 2 and equal to or more than the number of parameters. For example, for  $L_{16}(2^{15})$ , sixteen experiments could be used for analyzing two levels of 9-15 parameters.

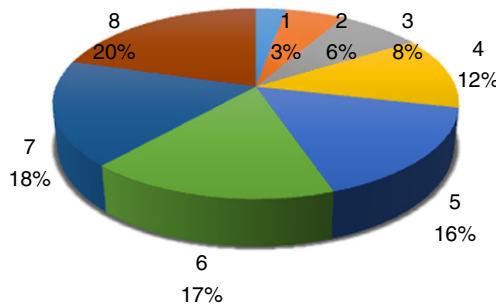


Figure 4. Roulette wheel used for chromosome selection

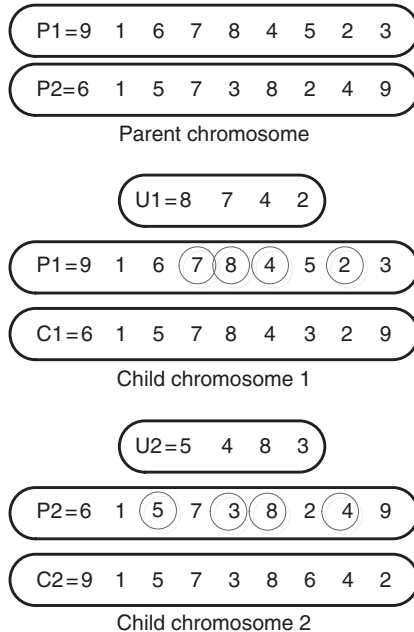
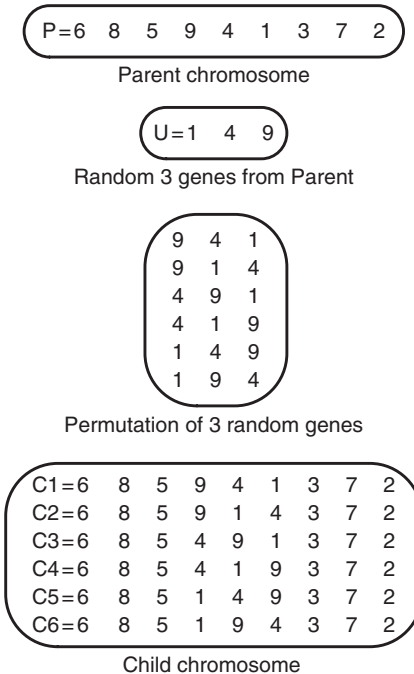


Figure 5. Set partition crossover

- Step 2 A pair of chromosomes from the crossover operation is randomly chosen, and each element of the pair is used to represent Level 1 and Level 2 in the OA.
- Step 3 The fitness value is determined using the OA  $L_n (2^{n-1})$  in  $n$  experiments. The number of chromosomes produced in this step is equal to the number of experiments.
- Step 4 The chromosome is evaluated by considering the SNR to either choose the best chromosome in the OA design or create a superior chromosome with a high SNR that is recommended by the OA. This step generates a single optimal chromosome.
- Step 5 The new superior chromosome produced by the Taguchi orthogonal array is added to the chromosomes produced in the crossover operation. The fitness function is then used for sorting the offspring.

5. Mutation: mutation is necessary as an exploration strategy to maintain diversity (Simon, 2013). In the current study, neighbor search mutation was used for the mutation operation. In this step, all the chromosomes obtained from crossover and the Taguchi OA are processed. Each chromosome is mutated if the random number is lower than the MR. Figure 6 shows three genes chosen randomly from a parental chromosome (U) with nine genes. In this figure, the permutation of U (containing [1 4 9]) leads to the creation of six (3!) sequences of genes. These six genes are used to create six offspring by using the parental chromosome, except the three genes generated from U1 to U6. Thus, this process generates six offspring chromosomes for each parental chromosome. Next, the six offspring are selected to choose the one with the best chromosome, based on their fitness value. Finally, the number of chromosomes produced in the mutation step is identical to that produced in crossover.



**Figure 6.**  
Neighbor search  
mutation

6. Elimination: in the elimination step, all chromosomes from parents and the chromosomes obtained in the previous step are combined, and NP chromosomes are chosen on the basis of the rank and fitness of all chromosomes. The processes from selection to elimination are performed in cycles until the target GN is reached.

#### 4. Experimental results and discussion

The objective of this study was to minimize the path along which the work table is moved to increase the efficiency of the drilling process. For examining the efficiency of the HTGA as a path optimizer, this study considered three problem cases. These three problem cases contained information on the coordinates of hole positions and were used in a control study. In a previous study, Lim *et al.* (2014) used these three problem cases for analyzing the efficiency of CS as a path drilling optimizer. The three problem cases, namely WP1, WP2, and WP3, have 9-, 14-, and 49-hole positions, respectively. Tables II-IV show the coordinates for each problem case.

The HTGA was embedded in a program code on Matlab R2012a software and run on a computer (Intel(R) Core(TM) i7-2600 CPU at 3.40 GHz and 8 GB RAM). Furthermore, the implementation and verification results of Lim *et al.* (2014) were used for comparing the results of the current study.

Table V shows the parameter setting in CS and the HTGA. Three parameters –  $p_a$ ,  $\alpha_c$ , and  $\alpha_h$  – were used only in CS. The CR and MR were used in the HTGA.

##### 4.1 Verification of WP1

WP1 has nine hole coordinates. The NP was set to 50, and the GN was set to 10,000. The HTGA was run 1,000 times.

Table VI shows the results of the HTGA for the optimal sequence and optimal time required in the CS case examined by Lim *et al.* (2014). For the remaining four criteria, CS reached convergence faster. By contrast, the HTGA has more processes for producing superior chromosomes, through selection, crossover, the use of the OA, and finally mutation. This problem case was too simple and easily solved by the HTGA, whereas a higher number of iterations was required for CS.

No.	X (mm)	Y (mm)	No.	X (mm)	Y (mm)
1	12.75	69.75	6	62.25	69.75
2	0.00	45.00	7	99.50	82.00
3	12.75	20.25	8	90.04	58.53
4	62.25	20.25	9	99.50	8.00
5	76.88	39.64			

**Table II.**  
Coordinates for  
problem case WP1  
(nine positions)

No.	X (mm)	Y (mm)	No.	X (mm)	Y (mm)
1	10.00	10.00	8	62.29	43.60
2	10.00	60.00	9	62.29	26.40
3	18.00	53.50	10	90.00	10.00
4	18.00	42.50	11	82.00	16.50
5	32.32	12.66	12	82.00	27.50
6	37.71	26.40	13	72.59	55.75
7	37.71	43.60	14	90.00	60.00

**Table III.**  
Coordinates for  
problem case  
WP2 (14 positions)

**Table IV.**  
Coordinates for  
problem case WP3  
(49 positions)

No.	X (mm)	Y (mm)	No.	X (mm)	Y (mm)
1	40.607	87.376	26	273.248	260.548
2	50.767	87.376	27	228.724	273.502
3	47.724	71.501	28	235.922	273.502
4	44.290	71.247	29	231.767	289.377
5	54.922	71.501	30	221.607	289.377
6	44.290	58.547	31	137.396	313.626
7	127.236	111.625	32	127.236	313.626
8	137.396	111.625	33	130.279	329.501
9	133.713	127.754	34	133.713	329.755
10	130.279	127.500	35	123.081	329.501
11	123.081	127.500	36	133.713	342.455
12	133.713	140.454	37	50.767	289.377
13	221.607	87.376	38	40.607	289.377
14	231.767	86.376	39	44.290	273.248
15	228.724	71.501	40	47.724	273.502
16	225.290	71.247	41	54.922	273.502
17	235.922	71.501	42	44.290	260.548
18	225.290	58.547	43	304.081	329.501
19	308.236	111.625	44	311.279	329.501
20	318.396	111.625	45	314.713	329.755
21	314.713	127.754	46	342.455	329.755
22	311.279	127.500	47	318.396	313.626
23	304.081	127.500	48	308.236	313.626
24	314.713	140.454	49	30.000	-5.000
25	225.29	260.548			

**Table V.**  
Parameter setting  
for CS and HTGA

Parameter	CS	HTGA
<i>For all WP</i>		
$p_a$	2/(number of holes-2)	na
$\alpha_c$	3/number of holes	na
$\alpha_h$	1	na
CR	na	0.6
MR	na	0.2
Run times	1,000	1,000
<i>For WP1 and WP2</i>		
PN	50	50
GN	10,000	10,000
<i>For WP3</i>		
PN	100	100
GN	20,000	20,000

Table VII shows that the HTGA obtained the same optimal solution as that obtained by CS. However, the HTGA requires a lower average number of iterations to converge, while requiring a higher number of iterations for global convergence. This result indicates that the HTGA outperforms CS in searching for a solution. The searching speed of these cases is shown in Figures 7 and 8.

4.2 Verification of WP2

WP2 consists of 14 hole positions. The NP was set to 50, and the GN was set to 10,000; the HTGA was run for 1,000 times.

WP2 is more complex than WP1. Thus, for this WP, the stability of the HTGA is more evident.

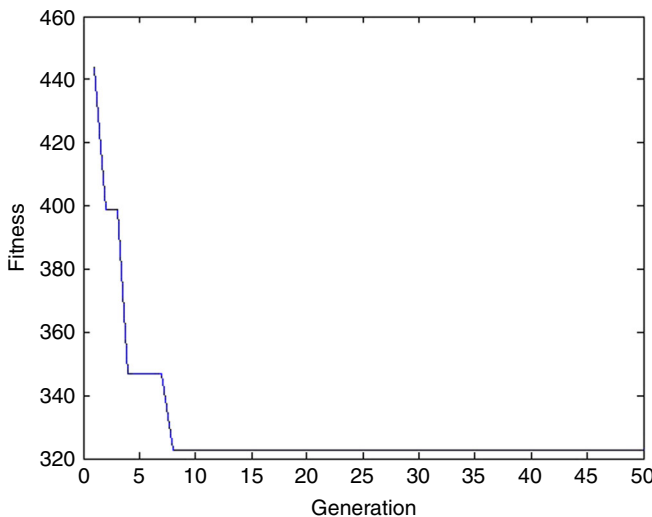
Tables VIII and IX show that the HTGA obtains the same optimal solution as CS obtains. Additionally, the HTGA shows superior performance in the lowest number of iterations, average number of iterations, and highest number of iterations during global convergence for both problem cases. The searching convergence of these problems case is shown in Figures 9 and 10.

Criteria	CS	HTGA
Optimal sequence	[3 2 1 6 7 8 5 4 9]	[3 2 1 6 7 8 5 4 9]
The least iteration number during global convergence	1	4
The average iteration number during global convergence	18	24
The most iteration number during global convergence	57	264
The optimal time required	322.5	322.5

**Table VI.**  
Comparison between  
CS and HTGA for  
WP1 (case 1)

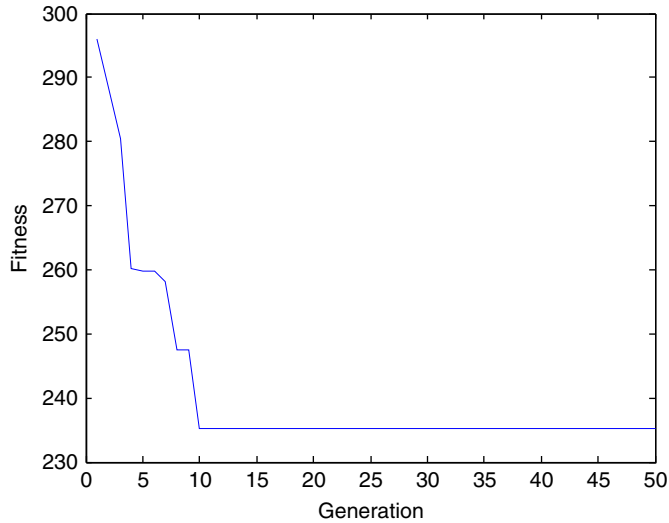
Criteria	CS	HTGA
Optimal sequence	[3 2 1 6 7 8 5 4 9]	[1 2 3 6 7 8 5 4 9]
The least iteration number during global convergence	1	1
The average iteration number during global convergence	13	10
The most iteration number during global convergence	40	33
The optimal time required	235.2	235.2

**Table VII.**  
Comparison between  
CS and HTGA for  
WP1 (case 2)



**Figure 7.**  
Graphic convergence  
results for WP1  
(case 1)

**Figure 8.**  
Graphic convergence  
results for WP1  
(case 2)



**Table VIII.**  
Comparison between  
CS and HTGA for  
WP2 (case 1)

Criteria	CS	HTGA
Optimal sequence	[10 11 12 9 6 5 1 2 3 4 7 8 13 14]	[1 5 6 9 12 11 10 14 13 8 7 4 3 2]
The least iteration number during global convergence	23	20
The average iteration number during global convergence	429	38
The most iteration number during global convergence	2,349	148
The optimal time required	280	280

**Table IX.**  
Comparison between  
CS and HTGA for  
WP2 (case 2)

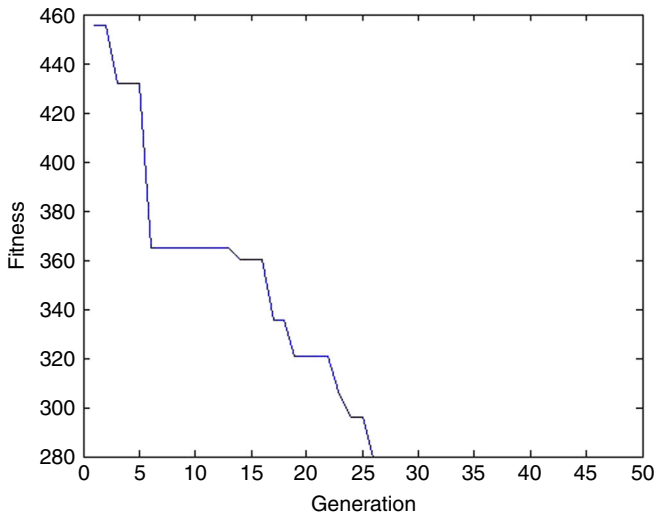
Criteria	CS	HTGA
Optimal sequence	[1 5 6 4 2 3 7 9 11 10 12 8 13 14]	[14 13 8 12 10 11 9 7 3 2 4 6 5 1]
The least iteration number during global convergence	30	18
The average iteration number during global convergence	375	86
The most iteration number during global convergence	2,092	343
The optimal time required	220	220

#### 4.3 Verification of WP3

WP3 contains 49 hole positions. The NP was set to 100, and the GN was set to 20,000; the HTGA was run 1,000 times.

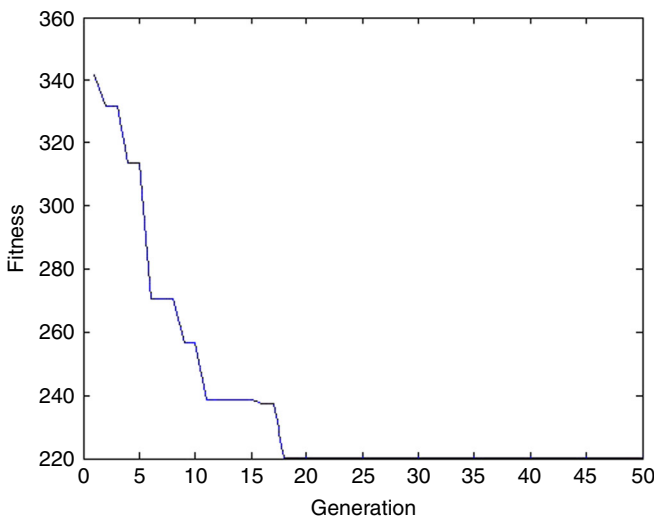
WP3 was the most complex problem in this study, and it has  $6.0828 \times 10^{62}$  (49!) feasible sequences. Therefore, the effectiveness of the HTGA was confirmed.

The second columns of Tables X and XI show the optimal sequences for WP3 in the study of Lim *et al.* (2014). In the two sequences, for Case 1, Gene 2 appears twice but Gene 42 is missing; for Case 2, Gene 6 appears twice but Gene 46 is missing. If the fitness functions of these sequences are obtained, a different result is derived



**Figure 9.**  
Graphic convergence  
results for WP2  
(case 1)

---



**Figure 10.**  
Graphic convergence  
results for WP2  
(case 2)

---

(the original results are written in brackets). To improve upon the poor result of Lim *et al.* (2014), two possible sequences were proposed for each case.

Tables X and XI show that the HTGA requires a shorter optimal time than CS does. However, for the remaining criteria, the HTGA cannot be compared with CS because of unavailable data.

According to the number of possible solutions and error percentage, the HTGA has been shown to be stable for obtaining the optimal solution. This performance is supported by many solutions generated through OA analysis. This stability improves the performance of the HTGA, thereby preventing it from being trapped in local optima.

**Table X.**  
Comparison between  
CS and HTGA for  
WP3 (case 1)

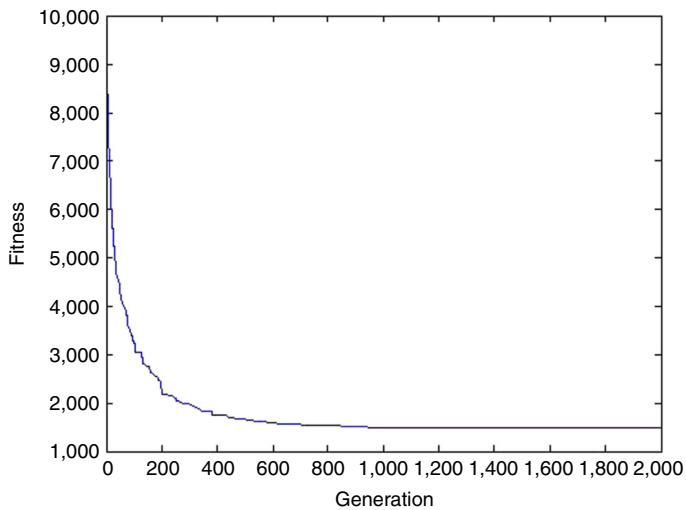
Criteria	Original CS	Feasible 1 CS	Feasible 2 CS	HTGA
Optimal sequence	[2 39 40 41 37 38 32 31 34 33 35 36 30 29 27 28 25 26 48 47 45 44 43 46 24 21 22 23 19 20 14 15 16 17 18 13 8 7 10 9 11 12 2 1 4 3 5 6 49]	[42 39 40 41 37 38 32 31 34 33 35 36 30 29 27 28 25 26 48 47 45 44 43 46 24 21 22 23 19 20 14 15 16 17 18 13 8 7 10 9 11 12 2 1 4 3 5 6 49]	[2 39 40 41 37 38 32 31 34 33 35 36 30 29 27 28 25 26 48 47 45 44 43 46 24 21 22 23 19 20 14 15 16 17 18 13 8 7 10 9 11 12 42 1 4 3 5 6 49]	[42 39 40 41 37 38 32 35 33 34 36 31 30 29 28 27 25 26 48 43 44 45 46 47 24 21 22 23 20 19 14 17 15 18 16 13 8 9 12 10 11 7 2 1 3 5 4 6 49]
The least time required	1,489.742	na	na	1,499.258
The average time required	1,805.262	na	na	1,531.017
The most time required	2,037.243	na	na	1,487.248
Percentage error for the least time required	2.02%	na	na	1.12%
Percentage error for average time required	23.63%	na	na	3.27%
Percentage error for most time required	39.52%	na	na	0.3%
The optimal time required	1,750.381 (1,460.216)	1,570.732	1,990.569	1,482.604

**Table XI.**  
Comparison between  
CS and HTGA for  
WP3 (case 2)

Criteria	Original CS	Feasible 1 CS	Feasible 2 CS	HTGA
Optimal sequence	[6 45 44 43 48 47 26 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 1 2 3 4 5 6 49 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[46 45 44 43 48 47 26 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 1 2 3 4 5 6 49 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[6 45 44 43 48 47 26 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 1 2 3 4 5 46 49 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[49 6 4 3 5 1 2 11 12 9 10 7 8 13 14 15 16 18 17 19 20 23 22 21 24 26 48 47 46 45 44 43 28 25 27 29 30 31 32 33 34 36 35 37 38 41 40 39 42]
The least time required	1,119.706	na	na	1,133.494
The average time required	1,338.65	na	na	1,163.198
The most time required	1,487.126	na	na	1,127.248
Percentage error for the least time required	1.06%	na	na	0.55%
Percentage error for average time required	20.82%	na	na	3.19%
Percentage error for most time required	34.23%	na	na	0%
The optimal time required	1,454.229 (1,107.927)	1,210.7630	2,000.016	1,127.248

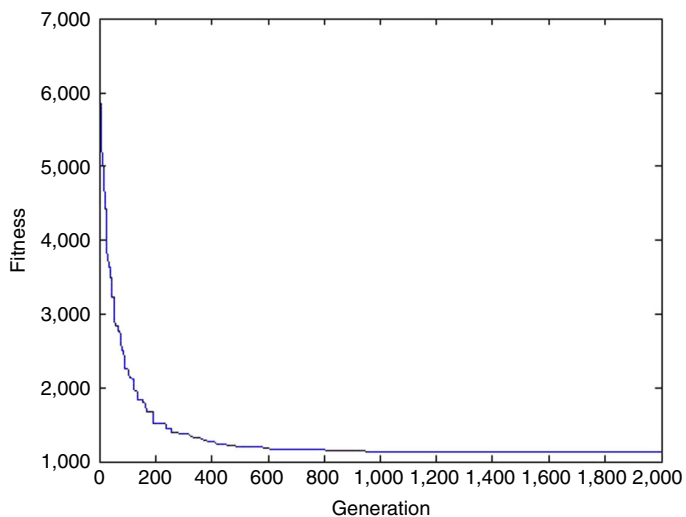
The aforementioned result indicates that the HTGA can be used and improved as an optimization method for complex problems. This statement is supported by the ability of the HTGA to analyze feasible solution candidates by using an OA. The combination of the HTGA and OA does not merely provide a solution; it provides the optimal alternative solution. The search performance of these problems case is presented in Figures 11 and 12.





**Figure 11.**  
Graphic convergence  
results for WP3  
(case 1)

---



**Figure 12.**  
Graphic convergence  
results for WP3  
(case 2)

---

## 5. Conclusion

The HTGA was used for optimizing a two-dimensional CNC PCB drilling path. Three classic TCPs examined by Lim *et al.* (2014) were considered to analyze the effectiveness of the HTGA. The procedure of the HTGA is similar to that of traditional GAs. The HTGA is a hybrid algorithm, and a Taguchi OA is introduced between crossover and mutation to improve chromosomes.

The results of the proposed algorithm were compared with those of the CS algorithm, which was used by Lim *et al.* (2014) for optimizing the drilling path, to verify the efficiency of proposed algorithm. The HTGA was more robust and showed faster convergence in reaching the optimal solution for WP2 and WP3. In particular, the HTGA obtained a superior solution for WP3, which was the most complex position problem in this study.

**References**

- Abbas, A.T., Hamza, K. and Aly, M.F. (2014), "CNC machining path planning optimization for circular hole patterns via a hybrid ant colony optimization approach", *Mechanical Engineering Research*, Vol. 4 No. 2, pp. 16-29.
- Al-Sahib, N.K.A. and Abdulrazzaq, H.F. (2014), "Tool path optimization of drilling sequence in CNC machine using genetic algorithm", *Innovative Systems Design and Engineering*, Vol. 5 No. 1, pp. 15-26.
- Fowlkes, W.Y. and Creveling, C.M. (1995), *Engineering Methods for Robust Product Design*, Addison-Wesley, MA.
- Gupta, A., Chandna, P. and Tandon, P. (2011), "Hybrid genetic algorithm for minimizing non productive machining time during 2.5 D milling", *International Journal of Engineering, Science and Technology*, Vol. 3 No. 1, pp. 183-190.
- Ismail, M., Othman, M., Sulaiman, H., Misran, M., Ramlee, R., Abidin, A., Nordin, N., Zakaria, M., Ayob, M. and Yakop, F. (2012), "Firefly algorithm for path optimization in PCB holes drilling process", *2012 International Conference on Green and Ubiquitous Technology (GUT), IEEE*, pp. 110-113.
- Lim, W.C.E., Kanagaraj, G. and Ponnambalam, S. (2012), "Cuckoo search algorithm for optimization of sequence in PCB holes drilling process", *Emerging Trends in Science, Engineering and Technology*, Springer, pp. 207-216.
- Lim, W.C.E., Kanagaraj, G. and Ponnambalam, S. (2014), "PCB drill path optimization by combinatorial cuckoo search algorithm", *The Scientific World Journal*, Vol. 2014, pp. 1-10.
- Mansour, G., Tsagaris, A. and Sagris, D. (2013), "CNC machining optimization by genetic algorithm using CAD based system", *International Journal of Modern Manufacturing Technologies*, Vol. 5 No. 1, pp. 76-80.
- Montiel-Ross, O., Medina-Rodriguez, N., Sepulveda, R. and Melin, P. (2012), "Methodology to optimize manufacturing time for a CNC using a high performance implementation of ACO", *International Journal of Advanced Robotic Systems*, Vol. 9, pp. 1-10.
- Omar, N., Eng Chieh, O., Adam, A., Hasim, S.H., Zainal Abidin, A.F., Jaafar, H.I., Zakaria, H., Jefery, H., Nordin, N.A. and Khairuddin, O. (2014), "An experimental study of the application of gravitational search algorithm in solving route optimization problem for holes drilling process", *International Conference Recent Trends in Engineering & Technology, Batam*, pp. 7-10.
- Qudeiri, J.E., Khadra, F.Y.A. and Al-Ahmari, A. (2013), "GA support system to optimize the sequence of multi-level and multi-tool operations in CNC machines", *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE*, pp. 231-236.
- Qudeiri, J.E.A., Raid, A.-M., Jamali, M.A. and Yamamoto, H. (2006), "Optimization hole-cutting operations sequence in CNC machine tools using GA", *2006 International Conference on Service Systems and Service Management, IEEE*, pp. 501-506.
- Sigl, S. and Mayer, H.A. (2005), "Hybrid evolutionary approaches to CNC drill route optimization", *International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, IEEE*, pp. 905-910.
- Simon, D. (2013), *Evolutionary Optimization Algorithms*, John Wiley & Sons, NJ.
- Tsai, J.-T., Liu, T.-K. and Chou, J.-H. (2004), "Hybrid taguchi-genetic algorithm for global numerical optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 8 No. 4, pp. 365-377.

- 
- Wei, W., Jian-Yong, L. and Heng, W. (2008), "Path optimization for PCB NC-drilling using genetic algorithm", *Computer Engineering and Applications*, Vol. 44, pp. 229-232.
- Yang, X.-S. (2010), *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Cambridge.
- Yang, X.-S. and Deb, S. (2009), "Cuckoo search via lévy flights", World Congress on Nature & Biologically Inspired Computing, 2009, IEEE, pp. 210-214.
- Ye, G. and Rui, X. (2013), "An improved simulated annealing and genetic algorithm for TSP", *2013 5th IEEE International Conference on Broadband Network & Multimedia Technology (IC-BNMT)*, IEEE, pp. 6-9.
- Zhu, G.-Y. and Zhang, W.-B. (2008), "Drilling path optimization by the particle swarm optimization algorithm with global convergence characteristics", *International Journal of Production Research*, Vol. 46 No. 8, pp. 2299-2311.

### About the authors

Dony Hidayat Al-Janan received the BS and MS Degree in Mechanical Engineering from the Muhammadiyah University of Surakarta, Indonesia in 2001 and Gadjahmada University, Yogyakarta, Indonesia in 2004, respectively. He is also a Lecturer of Computer Programming of Engineering Faculty, Semarang State University from 2006. Since 2013, he has been studying in a Doctoral Degree Program in Institute of Engineering Science and Technology, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, ROC, that supported by Indonesian Directorate General of Higher Education (DIKTI) scholarship in scheme BPPLN DIKTI 3+1. In his study, he interests to explore some method for optimization to simulate an appropriate method for particular problems.

Professor Tung-Kuan Liu received the BS Degree in Mechanical Engineering from the National Akita University, Akita, Japan, in 1992, and then the MS and PhD Degrees in Mechanical Engineering and Information Science from the National Tohoku University, Sendai, Japan, in 1994 and 1997, respectively. He is a Professor of Department of Mechanical and Automation Engineering and Graduate Program of Industrial Design, National Kaohsiung First University of Science and Technology, Taiwan ROC from February 2011. During October 1997 to July 1999, he was a Senior Manager with the Institute of Information Industry, Taipei, Taiwan. From August 1999 to July 2002, he was also an Assistant Professor with the Department of Marketing and Distribution Management, National Kaohsiung First University of Science and Technology. His research and teaching interests include artificial intelligence, applications of multi objective optimization genetic algorithms, and integrated manufacturing and business systems. Professor Tung-Kuan Liu is the corresponding author and can be contacted at: tkliu@nkfust.edu.tw