Int J Adv Manuf Technol (2017) 92:1263–1276
DOI 10.1007/s00170-017-0193-5

ORIGINAL ARTICLE

Tool path length optimisation of contour parallel milling based on modified ant colony optimisation

H. Abdullah [1] **· R . Ramli** [1] **· D. A. Wahab** [1]

Received: 2 December 2016 / Accepted: 20 February 2017 / Published online: 10 March 2017

Abstract Reducing machining time in a milling process is one of the important criteria to improve the overall efficiency of the machining process. This paper presents a study on the reduction of machining time, focusing on contour parallel machining to increase the efficiency and performance during the machining process. One method to enhance the performance of contour parallel machining is by defining a tool path interval that is larger than the radius of the cutting tool in a roughing operation because of its capability of reducing the tool path length and machining time. However, this causes the occurrence of an uncut region at the corner and at the centre of a contour parallel. This uncut region can be removed through an additional tool path known as the clear tool path. Therefore, in this paper, a new method based on an optimisation technique is introduced to generate a clear tool path that removes the entire uncut region in contour parallel machining at minimum cutting time. Ant colony algorithm (ACO) is used to optimise the clear tool path length in contour parallel machining time by minimising the movement of cutting tool in removing the entire uncut regions. A new transition rule has been established from the conventional ACO, which adapted the uncut region occurring at the corner of the contour parallel. Then, to validate the optimisation result, a cutting experiment was carried out using computer numerical control (CNC) milling machine. It can be ascertained from this study that the optimisation of the clear tool path gives optimal tool path length whilst reducing the cutting time in the roughing process.

Keywords Contour parallel · Milling · Uncut region · Ant colony optimisation · Computer numerical control

✉ R . Ramli
rizauddin@ukm.edu.my

H. Abdullah
haslinaa@uthm.edu.my

D. A. Wahab
dzuraidah@ukm.edu.my

[1] Department of Mechanical and Materials Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

1 Introduction

Pocket milling is a process in which all the materials inside a boundary profile are removed. It involves two processes: roughing and finishing. In the roughing process, materials are removed in large amounts, representing approximately 50% of the total machining time. It can be five to ten times longer than the finishing process [1]. Therefore, it is important to speed up the machining time during this process. One way to reduce the machining time during the roughing process is by decreasing the tool path length. Various factors that influence the tool path length include radial width, axial width, and techniques in the machining strategy, which are contour parallel and zigzag. Regarding machining time, the contour parallel method utilises a shorter time compared to the zigzag method [2]. However, in generating a contour parallel tool path, several factors need to be considered such as the tool path interval, connection between the contour segment method, and time consumption. The connection between the contour segments is significant as it influences the algorithm time consumed in generating the contour offset.

Due to this problem, many researchers have conducted studies to generate algorithms for contour parallel tool path. The existing literature reviews related to the generation of contour parallel tool path are based on pair-wise intersection

[3–5] and Voronoi diagram [6, 7]. In the pair-wise intersection approach, the tool path is generated through three steps: offsetting each contour edge, eliminating the self-intersection, and inserting a small arc to seal the gap between two adjacent offset edges, which removes the entire global invalid loop [3]. The main drawback of the pair-wise method, which involves the determination of all self-intersections and elimination of local invalid loops after an offset, is the longer time consumption [6]. The Voronoi method is constructed by dividing the surface into several segments based on the angle-bisector formula. This diagram offset approach is known to be efficient; however, it has numerical instability [3].

Kim et al. [8] proposed another method with algorithm based on the angle-bisector formula that can automatically eliminate the local invalid loop caused by the vertex offset. By employing this method, the issue of high consumption of time that occurs in the conventional pair-wise intersection approach can be solved. The algorithm consists of four steps: calculation of the offset vertices using bisectors, inspection of the validity using the invalid offset edge handling algorithm, elimination of global invalid loops, and combination of the generated boundaries and islands. However, due to the problem concerning the direction of the offset vertex, Lee et al. [9] improved the problem of handling the forward and backward edges, by imposing a segment to check the direction and position of the offset vertex. By applying this method, the existence of the local invalid loop can be prevented. Figure 1 shows the differences between the offsets of the edge and the vertex.

To improve the effectiveness of contour parallel machining, the tool path interval or radial width is defined to be larger than half of the cutting radius. This leads to an increase in the uncut regions at sharp corners, necks, and centres of the offset segment [1]. The question of the uncut region has been realised by many researchers, and to date, several research studies have been carried out to solve this problem. For instance, Choi and Kim [10] classified the uncut region that occurred in the contour parallel machining into three categories: corner, centre, and neck uncut regions as shown in Fig. 2.

These uncut regions can be removed by using an additional tool path known as clear tool path. In the study by Choi and Kim [10], the uncut region is discovered using a cutting simulation and removed by employing an additional clear tool path. Park and Choi [4] determined the presence of the uncut region by using the pair-wise intersection detection method. In this case, the uncut region was determined from the intersection between the outer tool envelope and the inner tool envelope. When it occurs, an additional tool path is added to remove the whole uncut region. In their study, the concept of interfering ranges, known as the Local Invalid Removal (LIR) and Global Invalid Removal (GIR), is introduced to improve the PWID method. Choy and Chan [11] also produced an additional tool path to remove the uncut region by proposing a single loop and a double loop of the tool path. By using this method, the entire uncut region occurring at the corner can be removed. In addition, the maximum thickness of the chip can be reduced. However, the addition of the tool path with two double loops increased the tool path length, which led to an increase in the machining time. Furthermore, the developed algorithm focussed solely on the problem of the uncut region occurring at the corner.

Therefore, Mansor et al. [12] applied the method of the Voronoi diagram to locate the uncut region and developed an additional tool path algorithm to discover all types of uncut regions at the corner, neck, and centre of contour parallel. In their proposed algorithm, the detection and removal of each type of uncut region were carried out by using the tool path interval, which can be increased to match with the diameter of the cutting tool. This increment reduced the tool path length and improved the machining time. In addition, the proposed algorithm removed the whole uncut region in all directions while generating a

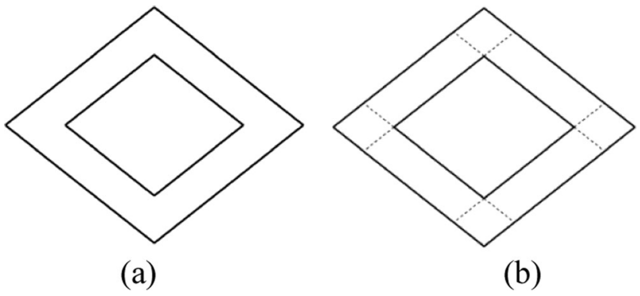


Fig. 1 Types of offset. **a** Vertex. **b** Edge

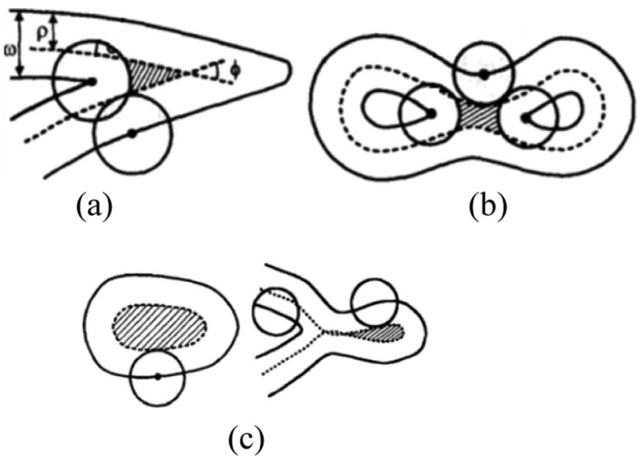


Fig. 2 Types of uncut region. **a** Corner, **b** neck, and **c** centre

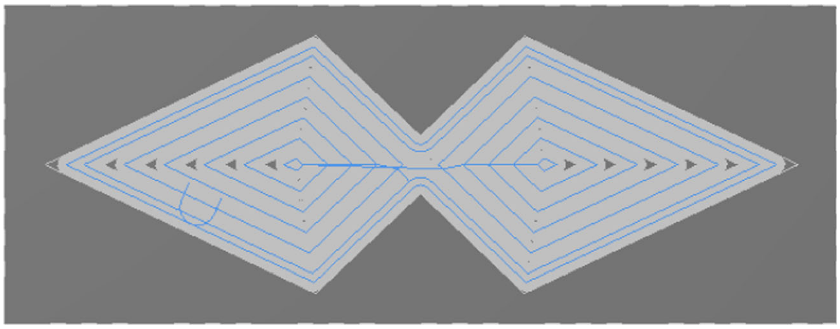


Fig. 3 Tool path and uncut region generated by Mastercam

tool path length that is shorter than the tool path produced using a computer-aided Manufacturing (CAM) software. However, there is a drawback in the cutting tool movement, which is the sudden change in the cutting tool direction. This is because for each type of the uncut region, more than one alternative or cutting path will be generated, which can cause a disarray in the movement of the cutting tool in eliminating the entire uncut region.

The latest study was conducted by Lin et al. [13] that proposed a clear tool path development based on geometrical analysis of the contour parallel offset. To generate a clear tool path, three steps were followed in the method. First is the detection of the uncut region as uncut lines. Then, these uncut lines are combined into several arcs based on the level, angle, and distance conditions. Finally, these arcs are further linked into a single curve according to centre parallel offset (CPO) centres. The single curve can be used as a clear tool path when a proper feed speed is set. Their proposed clear tool path has been proved by a cutting experiment. As a result, the algorithm generated a tool path that works accurately in the actual cutting conditions and can improve the machining efficiency of the roughing process. However, in order to join the uncut lines, the cutting tool should move several times on the same arc, and this can increase the tool path length and roughing time.

These methods are capable of removing the uncut region. However, they did not consider the movement of the cutting tool, which is producing a longer tool path and roughing time. Therefore, in this paper, an optimisation based on the artificial intelligence (AI) method has been carried out on the clear tool path to minimise the clear tool path length. A few studies using AI methods for minimising the tool path length have been conducted over the past. For example, Kumar et al. [14] and Gupta et al. [15] used genetic algorithm (GA) and hybrid genetic algorithm to optimise the non-productive machining time by minimising the non-productive tool path in contour parallel machining, while Oysu and Bingul [16] used the hybrid genetic algorithm and simulated annealing (SA) to minimise the non-productive tool path in contour parallel machining. Besides, Tian and Jiang [17] used ant colony optimisation (ACO) on pocket machining to create a minimum of cutting tool movement and reduce the non-productive machining time. In their study, the pocket was divided into sections based on the centre of contour parallel. Then, ACO was employed to determine the parts that had to be machined first by minimising the tool path length between the centres of contour parallel. Kiani et al. [18] also employed ACO to minimise the productive machining time on pocket machining with many types of geometry pockets in one workpiece. The ACO is used to determine the pocket that should be machined first to have a minimum cutting time. For each type of pocket, a coordinate in x-axis and y-axis will be defined as input in the ACO.

Each of the AI methods has its advantages and disadvantages. However, in the case of the path optimisation, the GA and ACO methods show an accurate and better result in terms

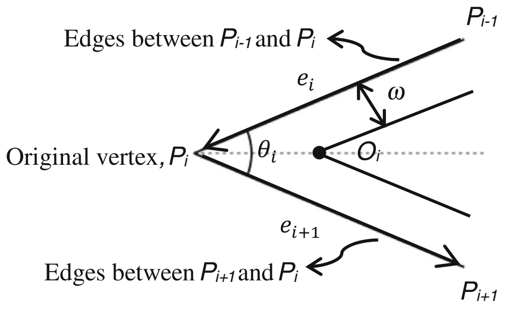
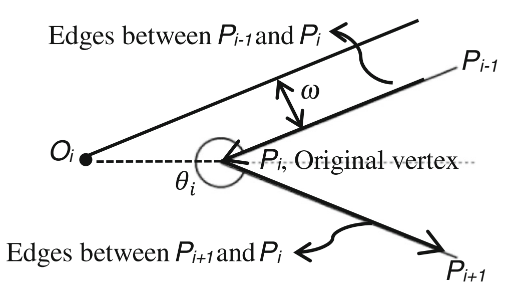


Fig. 4 Concave angle


Edges between P_{i-1} and P_i
P_{i-1}
ω
O_i
P_i, Original vertex
θ_i
Edges between P_{i+1} and P_i
P_{i+1}
Fig. 5 Convex angle

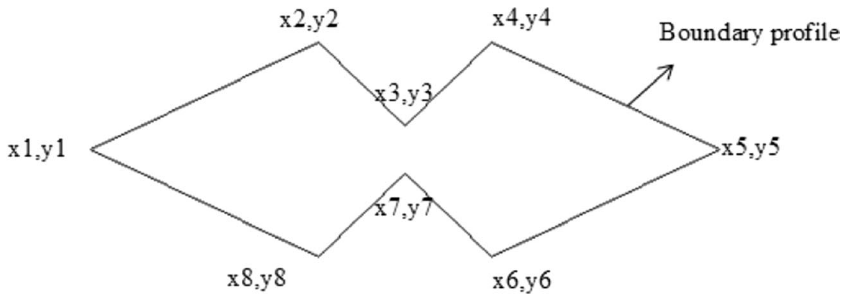


Fig. 6 Defining the coordinates of a boundary profile

of shorter route. For example, ACO has been used to optimise the tool path to determine the optimum path planning for computer numerical control (CNC) drilling machines for a special class of products that involve a large number of holes arranged in a rectangular matrix [19]. These optimisation techniques decreased the total travel distance of cutting tool as well as the machining time. Ghaiebi and Solimanpur [20] and Liu [21] also used the ACO method to optimise the process planning in drilling operation to minimise machining time, i.e. the air time and the tool-changing time. Rodríguez et al. [22] and Ross et al. [23] proposed a parallel ACO that can determine the best sequence of G commands of a set of holes for a printed circuit board. By using the parallel ACO, the hole-cutting sequence can shorten the cutting tool travel path. Furthermore, Abbas et al. [24] proposed a modified ACO for deciding the optimum tool path in the drilling process for products with a large number of holes arranged in concentric circular patterns. To study the effectiveness of the modified ACO, the obtained result has been compared to the conventional ACO and GA. As a result, the modified ACO can generate a shorter tool path length compared with conventional ACO and GA.

However, in this study, ACO is used as a method of optimisation due to the behaviour of the ant movement, and the algorithm can easily be adapted for the determination of the shortest route. Although the ACO has disadvantages in terms of algorithm convergence, compared with other methods, it can demonstrate better results due to the movement of ants that always ensure to find the shortest path in a short time [19, 25]. Therefore, the ACO was chosen as the optimisation method for determining the shortest clear tool path for removing the whole uncut region. The shortest tool path length is determined by optimising the movement of the cutting tool in removing the uncut region, which occurs at the corner and the centre of the contour parallel machining. For generating an optimal clear tool path length, three steps are followed: constructing contour parallel offset, determining the coordinate of the uncut region (expressed as an uncut line), and optimisation process based on ACO.

2 Offset algorithm and detection of uncut region

In this paper, the algorithm of contour parallel offset was developed to determine the coordinates of the uncut lines at every corner in each contour segment. The purpose of contour parallel construction is to generate a tool path that can be used in the actual machining process. In this study, the pattern of contour parallel offset was generated by using Matlab R2012b, which was translated from the contour parallel pattern produced by the CAM software, i.e. in this case, by using Mastercam. For instance, Fig. 3 shows the uncut region and the tool path of contour parallel generated by the CAM software based on the tool path interval, ω, is 5.7 mm. To ensure that the location of the uncut region generated by Mastercam matches the uncut region produced by the algorithm contour parallel offset, the pattern of contour parallel developed must be similar to the contour parallel as shown in Fig. 3.

In this paper, the offset of contour parallel was developed based on the method of angle bisector proposed by Kim et al. [8]. In this method, an offset point is produced based on the boundary profile, and local invalid loops are not created. Two types of angles are used: one that is smaller than 180°, π

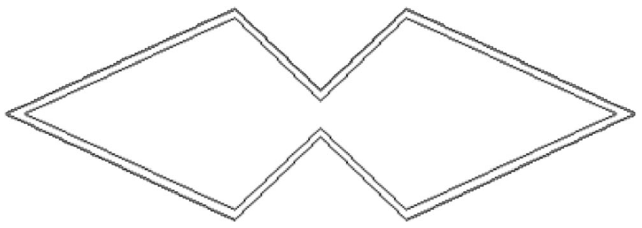


Fig. 7 First offset

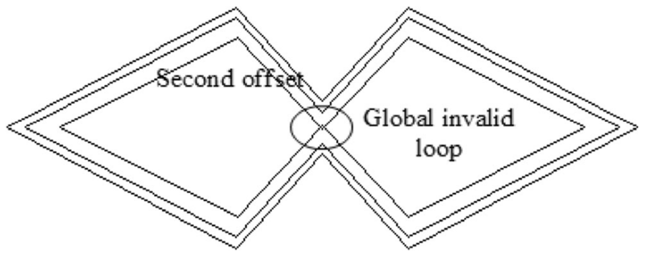


Fig. 8 Global invalid loop

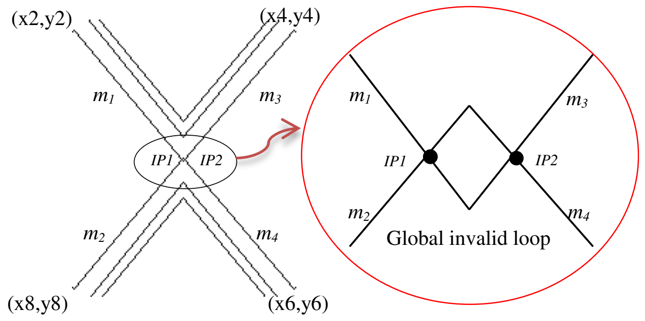


Fig. 9 Intersection point on global invalid loop

(0 < θ < 180°), known as a concave angle, and one that is larger than 180°, π (180° < θ < 360°), known as a convex angle, as shown in Figs. 4 and 5, respectively. For an angle smaller than 180°, the point offset is determined using Eq. 1; whereas for an angle larger than 180°, Eq. 2 is used. P_i is the original vertex between two original edges connected by vertex of P_{i-1}-P_i and P_i-P_{i+1}. The offset vertex O_i is calculated based on the original vertex P_i, the offset distance, which is also known as tool path interval (ω), the length of the two original edges (e_i, e_{i+1}), and the internal angle θ_i between the edges.

$$O_i = P_i + \frac{e_i - e_{i+1}}{|e_i - e_{i+1}|} \cdot \frac{\omega}{sin\left(\frac{\theta_i}{2}\right)} \tag{1}$$

$$O_i = P_i + \frac{e_i - e_{i+1}}{|e_i - e_{i+1}|} \cdot \frac{\omega}{cos\left(\frac{\pi - \theta_i}{2}\right)} \tag{2}$$

Where, e_i is presented by Eqs. 3 and 4:

$$e_i = \frac{P_i - P_{i-1}}{|P_i - P_{i-1}|} \tag{3}$$

$$e_{i+1} = \frac{P_{i+1} - P_i}{|P_{i+1} - P_i|} \tag{4}$$

To determine the offset point, the coordinates of the boundary's profile are defined as shown in Fig. 6. Next, by using the formula of the angle bisector, the offset point is determined. Furthermore, the direction and position of the offset point must be located within the profile boundaries so that the first offset can be generated as shown in Fig. 7. These steps are repeated to produce the next vertex offsets. However, the next vertex offset cannot be generated when a global invalid loop exists, as shown in Fig. 8.

Thus, to continue with the next offset, the invalid global loop must be eliminated first by computing the coordinates at the intersection point (IP) in the area of the global invalid loop. The IP can be acquired by determining the slope, m, for each line that intersects both vertexes. Figure 9 shows the intersection point on the global invalid loop.
where.

x,y	coordinates on profile offset
m	slope of edges
IP1	intersection point 1
IP2	intersection point 2

The slope, m, can be determined by using Eq. 5:

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_1} \tag{5}$$

Once the coordinates of the intersection points are obtained, the offset vertex is built based on this intersection, as shown in Fig. 10. Then, the subsequent process offsets

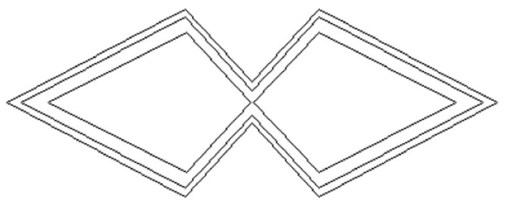


Fig. 10 Offset after elimination of the global invalid loop

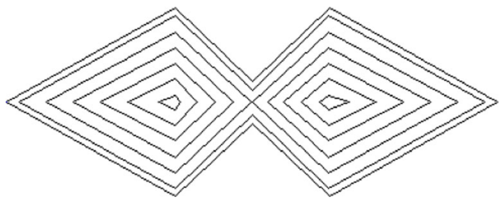


Fig. 11 Complete contour parallel offset

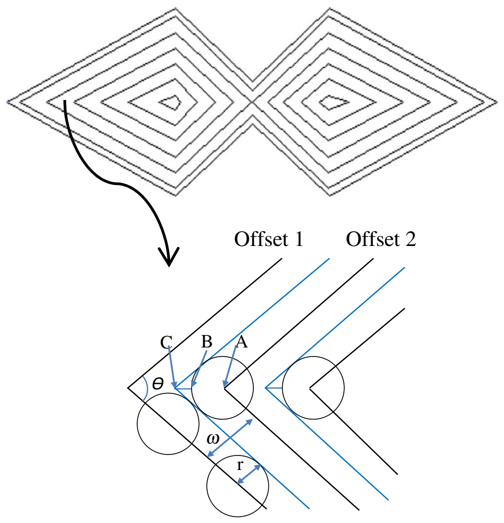


Fig. 12 Uncut region at the corner

```
for point=1:n (number of points)
        for i=1:m (number of contours)
                A(n)=(x y) (offset point on each contour)
                B=A+r*e;
                B1C1 =((2*r-r)/sin (angle1/2))-r;
                C=B+|BC|*e;
        end
end
```

Fig. 13 Algorithm for determination of uncut region

are generated until the end of the contour segment. To produce a new offset, the previous contour segment is defined as the boundary profile. These steps continue until the production of a complete contour parallel offset, as shown in Fig. 11.

After a complete offset is produced, the detection of the uncut region was carried out by analysing the contour parallel. Each uncut region on the segment contour is represented by an uncut line, which is defined on the *x*- and *y*-axes. The uncut region is represented by two points that are from point B to point C, as shown in Fig. 12. For the corner, if the path interval, ω, fulfils the following condition in Eq. 6, the uncut regions will be left at the corner.

$$\omega > r\left(1 + sin\left(\frac{\theta}{2}\right)\right) \tag{6}$$

Hence, the uncut region, which is point B and point C, can be determined by:

$$B = A + r * e \tag{7}$$

$$C = B + |BC| * e \tag{8}$$

Where:

$$|BC| = ((\omega - r)/sin(\theta/2)) - r \tag{9}$$

Where, *e* denotes the unit vector of the angular bisector AC directing from point A to point C. The algorithm for the determination of the uncut region is shown in Fig. 13, and the example of the entire uncut region determined in the contour parallel is shown in Fig. 14.

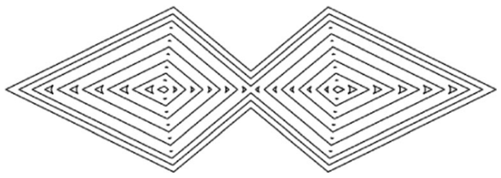


Fig. 14 Uncut region in contour parallel

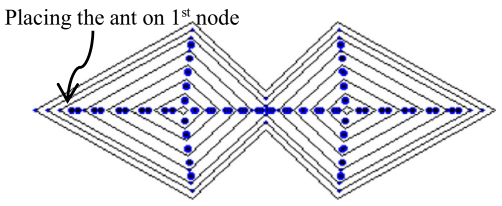


Fig. 15 Placing of ant on the first node

3 Ant colony optimisation

Ant colony optimisation is a meta-heuristic method that is adopted from the ant behaviour, which is seen in ant colonies. This was introduced by Dorigo et al. [26]. Ants have the ability to sense a complex environment while searching for food, and when they return to the nest, they leave a pheromone substance on the routes that they follow. The ants communicate with each other by using the pheromone trail left in the path. From this, the other ants will choose the shortest path left by the pheromone trail. Higher evaporation will occur along the path that is less chosen by the ants and that will reduce the effect of the pheromone trail. On the contrary, the paths that are frequently used by the ants will leave a stronger pheromone trail and eventually the route will be chosen as the shortest path to return to the nest. In this study, the ant movement from one node to another represents the movement of the cutting tool from the first uncut region to another uncut region. At each iteration, all the ants will be placed at the first point of the uncut region, as shown in Fig. 15.

Then, the movement of ants to the next node will be determined based on a probability rule as in Eq. 10.

$$P^k_{i,j}(t) = \frac{\left[\tau_{i,j(t)}\right]^{\alpha}\left[\eta_{i,j}(t)\right]^{\beta}}{\sum_{t\epsilon N^k_i}\left[\tau_{i,j}(t)\right]^{\alpha}\left[\eta_{i,j}(t)\right]^{\beta}}\ j\epsilon N^k_i \tag{10}$$

Where:

N^k_i	list of nodes that have not been visited by ant *k*
$\tau_{i,j}(t)$	intensity of trail on the edge (*i,j*) at time *t*
α	weight of the pheromone

```
Set parameters;
For i=1: iteration
        Ant primary placing;
        Construct ant solution and ant cost;
        Updates the pheromone trail;
end
```

Fig. 16 Pseudo code of ACO algorithm

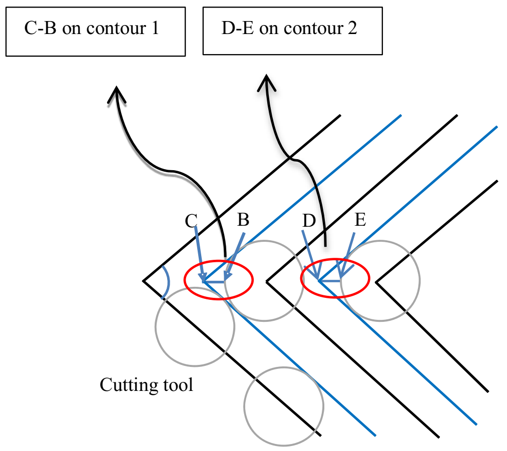


Fig. 17 Uncut line in each contour

$\eta_{i,j}(t)$	$1/d_{ij}$ is the visibility
β	weight of the visibility

The selection of the next node is based on the weight of the pheromone and weight of visibility. τ_{ij} is the pheromone value on the edge between node i and node j; whereas, $1/d_{ij}$ is the inverse distance of the movement of ants between the two nodes. d_{ij} is determined based on the Euclidean equation as in Eq. 11.

$$d_{ij} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2} \tag{11}$$

At every iteration, each node that has been visited by the ants is listed in a table called Tabu list. In this way, it can be ensured that the ants have not passed through the same point again. After all ants visit the uncut region, they will return to the first node, and the level of pheromone on each edge will be updated based on the local pheromone update rules as in Eqs. 12 and 13.

$$\tau\ (i,j) = (1-\rho)\ \tau\ (i,j) + \textstyle\sum_{k=1}^{m} \Delta\tau_k\ (i,j) \tag{12}$$

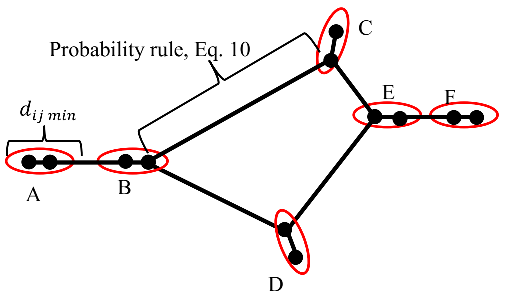


Fig. 18 Selection of nodes based on modified ACO

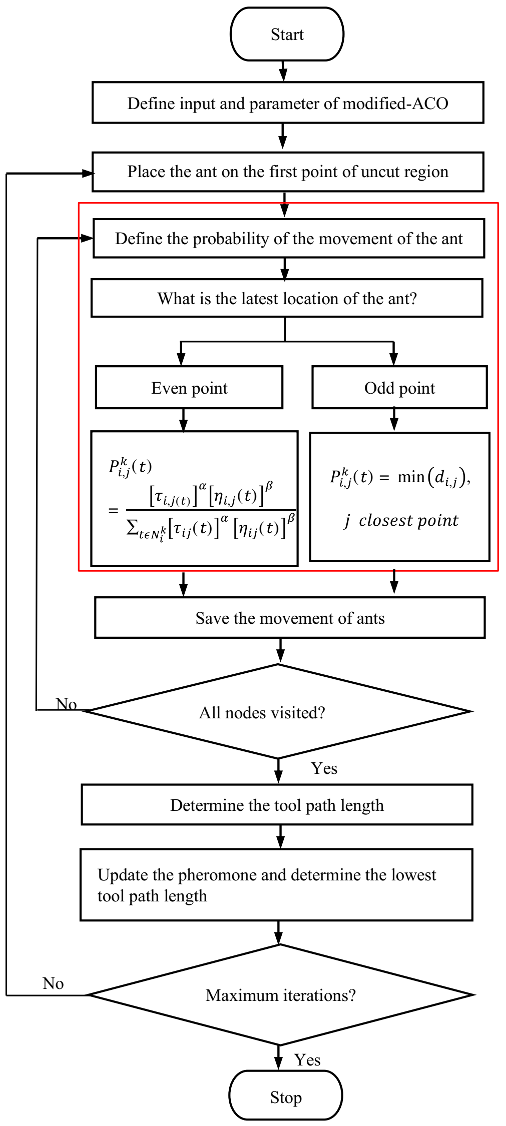

Start
Define input and parameter of modified-ACO
Place the ant on the first point of uncut region
Define the probability of the movement of the ant
What is the latest location of the ant?
Even point
Odd point
$P_{i,j}^k(t) = \frac{[\tau_{i,j(t)}]^\alpha [\eta_{i,j}(t)]^\beta}{\sum_{t\epsilon N_i^k}[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}$
$P_{i,j}^k(t) = \min(d_{i,j}),$
j closest point
Save the movement of ants
All nodes visited?
No
Yes
Determine the tool path length
Update the pheromone and determine the lowest tool path length
Maximum iterations?
No
Yes
Stop

Fig. 19 Flow chart of modified ACO

Table 1 Combination of α and β for conventional ACO

β \ α	0	1	2	3	4	5
0	○	○	○	○	○	○
1	○	○	◍	◍	◍	◍
2	◍	◍	◍	◍	●	◍
3	◍	◍	◍	●	●	●
4	●	●	●	●	●	●
5	●	●	●	●	●	●

●	Good combination
◍	Poor combination
○	Bad combination

Table 2 Combination of α and β for modified ACO

β \ α	0	1	2	3	4	5
0	○	○	○	○	○	○
1	○	◍	◍	◍	◍	◍
2	○	◍	◍	◍	●	●
3	◍	◍	◍	●	●	●
4	◍	◍	●	●	●	●
5	●	●	●	●	●	●

● Good combination
◍ Poor combination
○ Bad combination

Where,

$$\Delta\tau_k\,(i,j) = \begin{cases} 1/L_k & \text{if } (i,j) \in \text{ journey } by\ ant\ k \\ 0 & \text{others} \end{cases} \tag{13}$$

L_k is the distance of the movement of the ant starting from the first node of the uncut region until all the uncut regions are machined. In other words, this is the tool path length of the cutting tool for removing all the uncut regions that occur in the contour parallel machining. L_k is also known as the cost function, which should be minimised in the ACO. The evaporation rate, ρ, is a parameter that takes the value between 0 and 1. Figure 16 shows the pseudo code of the ACO algorithm that was used for this study.

4 Modified ant colony optimisation

In this paper, the ACO algorithm was used to find the solution for the uncut region problem in contour parallel machining. In this, all the ants were placed on the first node of the uncut region and then moved to the next node based on the probability rule, as shown in Eq. 10. The probability rule relies on the weight of the pheromone and the inverse distance. Then, a random number was generated, and the movement of the ants to the next node was determined using the roulette wheel method. However, by using the selection method based on the probability rule and roulette wheel, if the cutting tool is on the first point of the uncut region (point C), there is a possibility that the cutting tool has not been through the second point (point B) of the uncut region. Meanwhile, each uncut region in the segment contour is represented by a line consisting of two points (C–B), as shown in Fig. 17. Therefore, in this case, if the ants are placed on the first node at point C, they are required to move to the closest node, which is the second node at point B, so that the uncut region on contour 1 can be eliminated completely. Thus, new transit rules for ACO were developed to ensure that the tool can machine the whole surplus area in each segment of the contour before moving to the next contour.

In the modified version of ACO, there is a difference in the method of determining the movement of ants to the next node. The movement of the ants is dependent on the new transit rules as in Eq. 14.

$$New\_P_{ij} = \begin{cases} d_{ijmin} & \text{if } i = 1st \text{ node uncut region } at\ i \text{ contour} \\ \dfrac{\left[\tau_{i,j(t)}\right]^{\alpha}\left[\eta_{i,j}(t)\right]^{\beta}}{\sum_{t\epsilon N_i^k}\left[\tau_{i,j}(t)\right]^{\alpha}\left[\eta_{i,j}(t)\right]^{\beta}} & j\epsilon N_i^k \end{cases} \tag{14}$$

The movement of ants relies on two conditions, which is the minimum distance between the two nodes, d_{ijmin}, and a probability function shown in Eq. 10. If the cutting

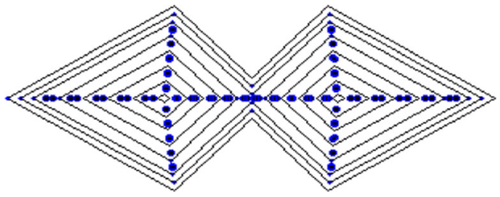


Fig. 21 Location of the uncut region at the corner

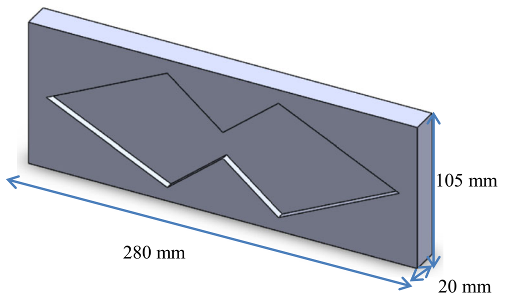


Fig. 20 First model

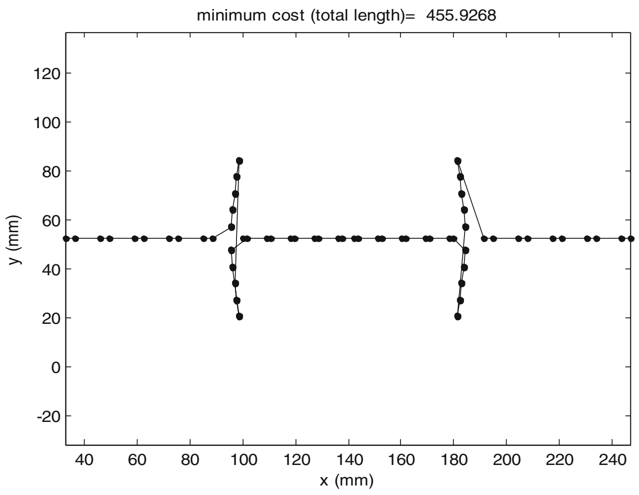


Fig. 22 Clear tool path length based on modified ACO

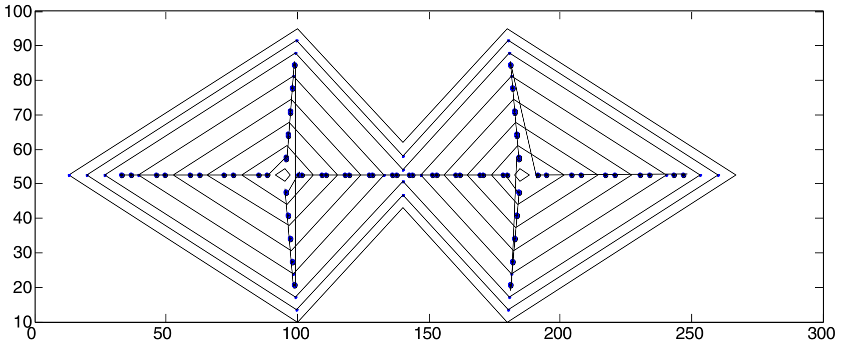


Fig. 23 Total tool path generation based on modified ACO

tool is located at the first node of the uncut region, then it will move to the closest node at the second point on the same uncut region. This ensures that the cutting tool removes the whole uncut region before moving to the next contour. The movement of the cutting tool from the last node of the uncut region is dependent on the probability rule as in Eq. 10. For example, Fig. 18 shows the selection of movement of the cutting tool from the uncut region A till the uncut region F. If the cutting tool is at the first node of the uncut regions A, B, C, D, E, or F, d_{ijmin} is used to ensure that the cutting tool moves to the closest node. The minimum distance was determined based on the Euclidean in Eq. 8. To determine the movement of the cutting tool between uncut regions, Eq. 10 was used.

By employing the conventional ACO, if there are 20 nodes, there will be 6×10^{16} solution of routes; however, by utilising the ACO with new transit rules, half of the nodes are determined based on the minimum distance between two nodes and the other half are based on the probability rule. This will reduce the number of solutions in the simulation process. In this way, it can speed up the process and reduce the simulation time in determining the most optimal tool path length. The summary of the modified ACO is depicted as a flow chart, as shown in Fig. 19.

5 Parameter setting in ACO and modified ACO

Some parameters need to be defined in ACO before running the simulation. In this study, the parameter setting is defined based on the experimental fixed setting method proposed by Dorigo et al. [26] and Luo et al. [27]. The parameters are defined based on the probability rule and local pheromone update. For conventional ACO and modified ACO, the two parameters defined according to the Experimental Fixed Setting are the weight of trail, α, and weight of visibility, β. As introduced previously, parameters α and β influenced the selection probability of ant movement. Therefore, ten simulation runs were performed for each combination to ensure that the best combination of these parameters is chosen to generate the optimal tool path length. As illustrated in Tables 1 and 2, the best solution can only be found in certain sets of combinations of α and β. The good combination of α and β gives an algorithm that can produce a shorter tool path length. This combination is suggested to be defined as the parameter values before running the simulation. In this study, the chosen combinations of α and β are 4 and 5, respectively, for ACO and modified ACO. By using this combination, the shortest tool path length can be obtained. The number of ants, m, and the evaporation rate, ρ, have been defined as suggested by Dorigo et al. [26].

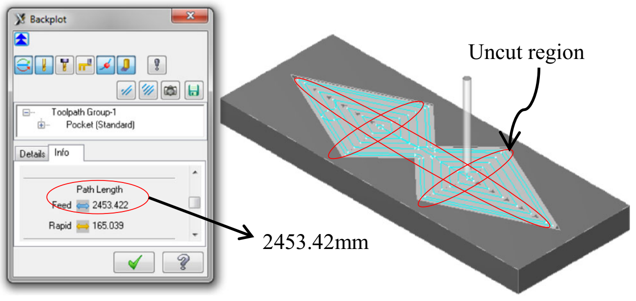


Fig. 24 Length of contour parallel, ω = 5.7 mm

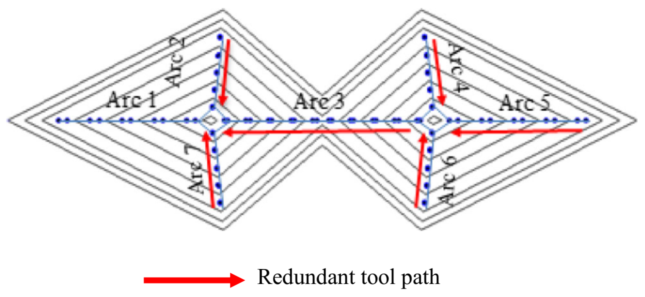


Fig. 25 Clear tool path proposed by Lin et al. [13]

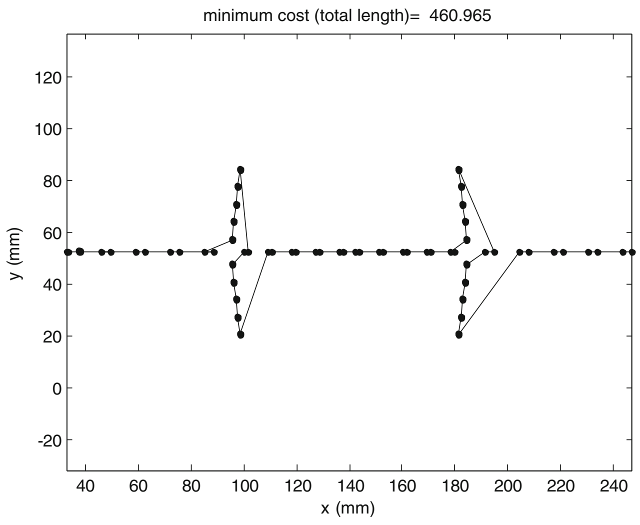


Fig. 26 Clear tool path based on conventional ACO

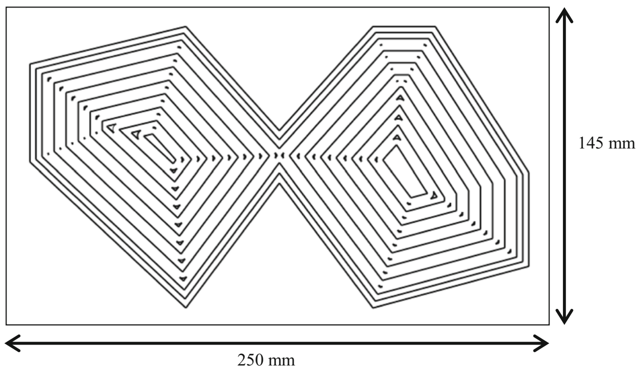


Fig. 28 Second model

6 Implementation of the algorithm

The algorithm of contour parallel offset and ACO algorithm were developed by using Matlab R2012b. The algorithm of the conventional ACO was tested on a simple model by Abdullah et al. [28]. The obtained result has been compared with that of the research by Lin et al. [13]. The simulation results show a reduction of 5% for machining time. In this present paper, our proposed modified ACO has been tested on two models in order to observe the optimisation efficiency when generating the clear tool path. The geometry of the first model to test this algorithm is similar to the model in the previous journal by Lin et al. [13], as shown in Fig. 20. The model is made of aluminium A6061.

In order to establish the coordinate of the uncut line, the contour parallel offset was constructed using the algorithm shown in Fig. 13. The cutting tool size used for this study is 6 mm and the tool path interval, ω, is 5.7 mm so as to make sure that the uncut region occurs at the corner. Each coordinate of the corner at the contour segment was defined, and the result is illustrated in Fig. 21. These coordinates are set as an input to our modified ACO.

In the modified ACO, the parameters of α and β were set as 4 and 5, respectively. Figure 22 shows the result of the optimal tool path length of 455.93 mm. This tool path length refers to the complete movement of the cutting tool, right from removing the first until the last uncut region which is known as the clear tool path. Consequently, Fig. 23 shows the entire movement of the cutting tool for the roughing process which consists of the tool path of contour parallel machining and the clear tool path.

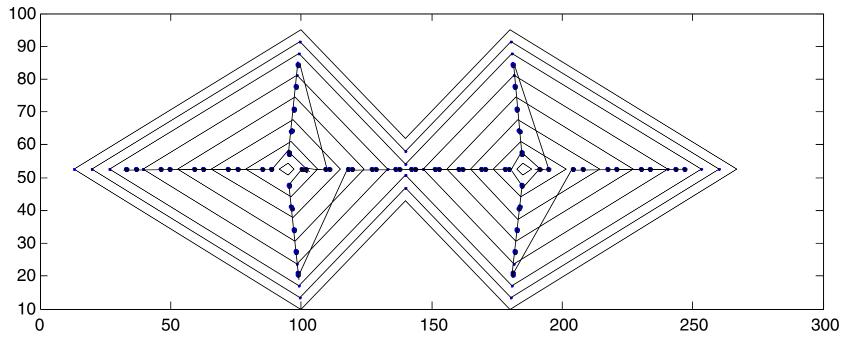
In order to determine the machining time (T_m) in the roughing process, Eqs. 15 and 16 were used [15]. Based on Eq. 15, the machining time depends on the tool path length and feed rate (f).

$$T_m = \frac{L_T}{f} \tag{15}$$

$$L_t = L_{cp} + L_{ct} \tag{16}$$

L_t is the complete tool path length during the roughing process, consisting of the length of contour parallel, L_{cp},

Fig. 27 Total tool path generation based on modified ACO



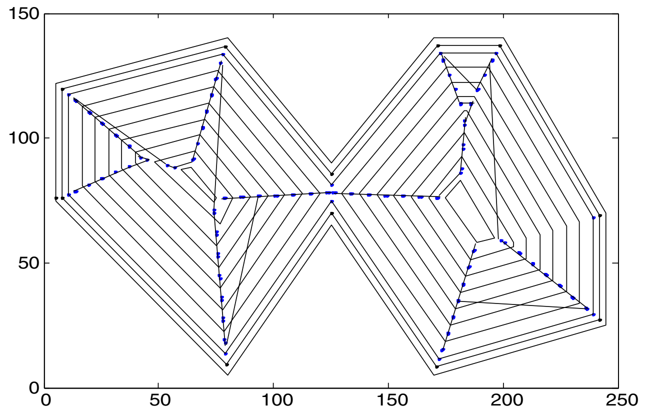


Fig. 29 Tool path strategies for a complex model based on ACO

and length of the clear tool path, L_{ct}, which was obtained based on ant colony optimisation. The length of the contour parallel is referred to as the length before the uncut region is removed and can be defined directly from the Mastercam software as depicted in Fig. 24, which is 2453.42 mm. By using Eq. 15, the roughing time obtained with the modified ACO is 291 s.

To determine the effectiveness of optimisation for the clear tool path, the obtained result was compared to the result of the method proposed by Lin et al. [13]. Figure 25 shows the tool path generated using an analysis of geometry for contour parallel. The time of the roughing process produced is 312 s, which is 7.2% higher than the time obtained using the optimisation method. It is proven from the results that by using the optimisation process, a shorter clear tool path is produced. This is due to the elimination of the redundant tool path that occurred when the cutting tool returned to the centre of the contour parallel as illustrated in Fig. 25. As a result, the modified ACO has reduced the tool path length in the roughing process and it can speed up the machining time.

Additionally, simulation has been carried out on the first model by using the conventional ACO in order to observe the effectiveness of the new transition rule. Figure 26 shows the optimal length of the clear tool path obtained from the conventional ACO, which is 460.9 mm. Meanwhile, the complete tool path consists of the contour parallel machining and the clear tool path generated based on the modified ACO as shown in Fig. 27. Similar to the optimisation based on modified ACO, the time of the roughing process was also determined using Eqs. 15 and 16; the tool path length of the contour parallel was 2453.42 mm. The roughing time produced by the conventional ACO is 293 s. Therefore, it is proved that by using the new transition rule, the clear tool path length can be reduced by 1.2%. In addition, by using the modified ACO, it can be ascertained that the cutting tool has removed the uncut region on each contour before moving to another segment of the contour. When the cutting tool reaches the centre of the contour line, it moves to the closest uncut region. In other words, the movement of the cutting tool in removing the whole uncut region in a roughing process based on our modified ACO is more structured and efficient.

Another example to test the proposed algorithm is shown in Fig. 28. The outer profile is based on a modification to an example model obtained from a previous research by Kim [29]. The purpose of this model is to identify the effect of optimisation on more complex models that have more uncut regions. Figures 29 and 30 show the result of the tool path generated based on ACO and modified ACO, respectively. The clear tool path lengths obtained by ACO and modified ACO are 749.76 and 728.92 mm, correspondingly. By using Eqs. 15 and 16, the times of roughing process obtained were 515 and 512 s for ACO and modified ACO, respectively. On comparing to the machining time acquired from the analysis on contour parallel proposed by Lin et al. [13], the ACO managed to reduce the roughing time by 3.5%. Meanwhile, by using the modified ACO, the time of roughing process was reduced by 4.2% compared with the analysis on contour parallel.

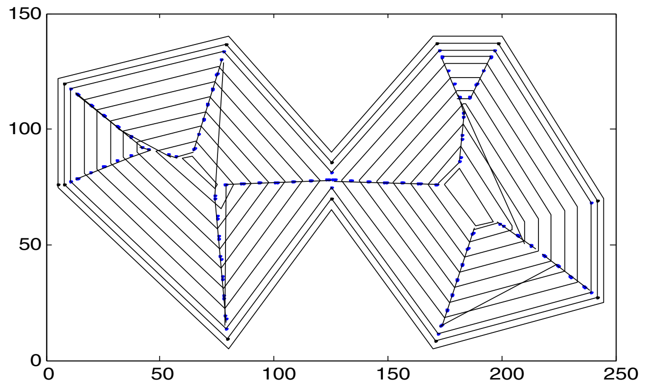
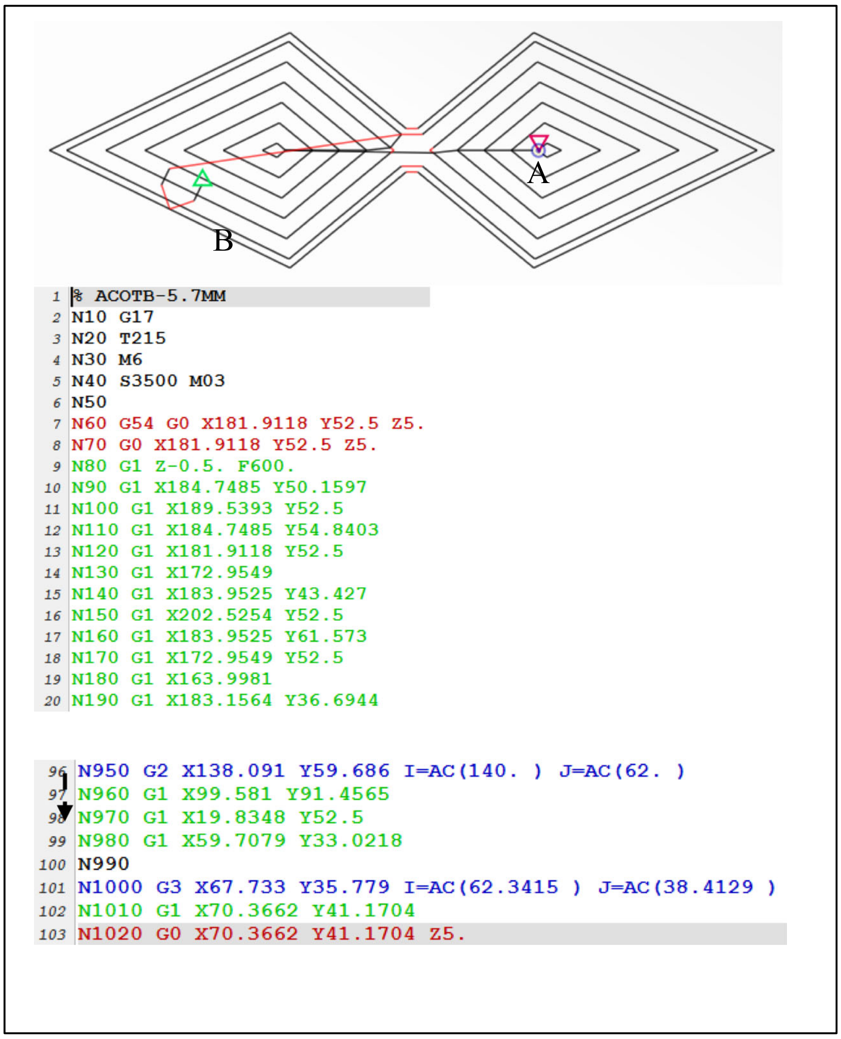


Fig. 30 Tool path strategies for a complex model based on modified ACO

7 Experiments

Experiments were carried out to validate the results obtained from the optimisation and to determine whether the proposed clear tool path based on the modified ACO can be practised in actual machining. In addition, the tests were conducted to ascertain whether the optimised tool path can improve the method proposed by Lin et al. [13]. For the machining process, a flat-end mill-cutting tool with a diameter of 6 mm has been used to cut the material made of aluminium A6061. The

Fig. 31 First phase of G-code



```
% ACOTB-5.7MM
N10 G17
N20 T215
N30 M6
N40 S3500 M03
N50
N60 G54 G0 X181.9118 Y52.5 Z5.
N70 G0 X181.9118 Y52.5 Z5.
N80 G1 Z-0.5. F600.
N90 G1 X184.7485 Y50.1597
N100 G1 X189.5393 Y52.5
N110 G1 X184.7485 Y54.8403
N120 G1 X181.9118 Y52.5
N130 G1 X172.9549
N140 G1 X183.9525 Y43.427
N150 G1 X202.5254 Y52.5
N160 G1 X183.9525 Y61.573
N170 G1 X172.9549 Y52.5
N180 G1 X163.9981
N190 G1 X183.1564 Y36.6944

N950 G2 X138.091 Y59.686 I=AC(140. ) J=AC(62. )
N960 G1 X99.581 Y91.4565
N970 G1 X19.8348 Y52.5
N980 G1 X59.7079 Y33.0218
N990
N1000 G3 X67.733 Y35.779 I=AC(62.3415 ) J=AC(38.4129 )
N1010 G1 X70.3662 Y41.1704
N1020 G0 X70.3662 Y41.1704 Z5.
```

cutting speed was set as 600 mm/min as suggested by Lin et al. [13]. However, since the rapid movement for the distance between the uncut regions is larger than the length of the arc, the cutting speed was set as 10,000 mm/min because no uncut region was removed. The axial depth was maintained for the entire roughing process, which was 0.5 mm. The model was created using SOLIDWORKS and saved as STL file and then imported to Mastercam to generate and visualise the tool path. The processor of MPSIEM_A.MMD was used to generate and transfer the G-code to the CNC three-axis milling machine of type Hermle UWF 900. In this study, two phases were

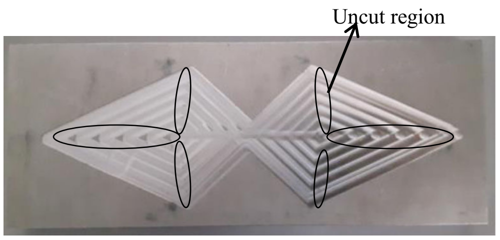


Fig. 32 Uncut region occurs when $\omega = 5.7$ mm

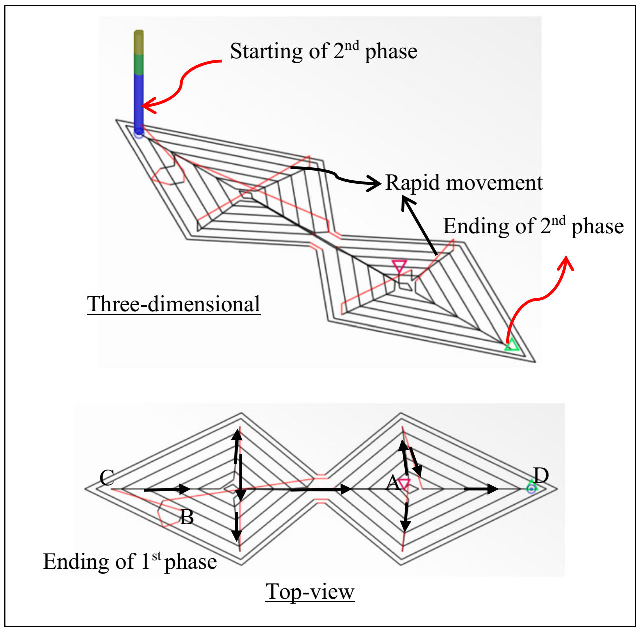


Fig. 33 Global tool path when $\omega = 5.7$ mm

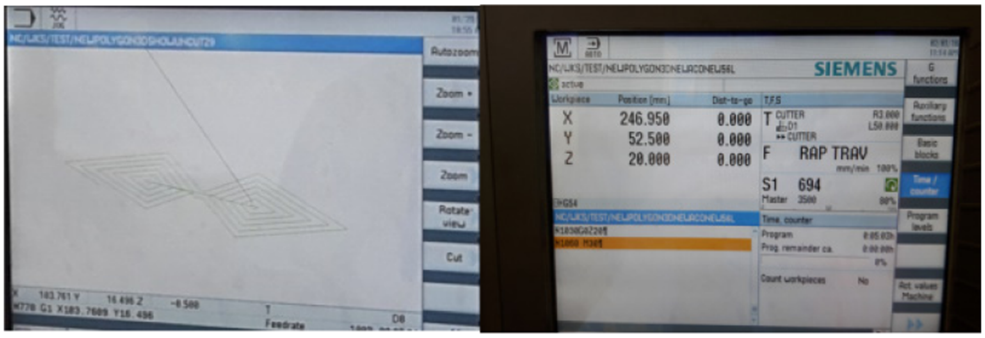


Fig. 34 Machine controller. **a** Checking process. **b** Cutting time record

involved in generating the G-code. Phase 1 was the generation of G-code automatically in Mastercam because of the contour parallel tool path. Phase 2 is the generation of G-code that will be used to remove the uncut region in contour parallel machining. The coordinate of G-code in Phase 2 is determined based on the movement of the cutting tool obtained from the ACO process.

Figure 31 shows the generated G-code of Phase 1 based on the contour parallel tool path. The cutting tool started at point A and then moved to the next contour until it reached point B. Figure 32 shows the uncut region at the corner based on the actual machining process. In order to remove the entire uncut region, the G-code in Phase 2 was created using the optimal tool path length obtained from the modified ACO. Consequently, Fig. 33 shows that once the cutting tool ended at point B, it continued to move to the next point C which is the first point of the uncut region. This was the beginning of G-code generation of Phase 2. Then, the cutting tool moved to cut the entire uncut region based on the optimal tool path as shown in Fig. 33 (top view). The roughing process for the axial depth of 0.5 mm ended at point D.

After G-code generation is completed, it was transferred into the CNC milling machine and stored in the machine controller as shown in Fig. 34a. In this step, the tool path was checked to ensure that the generated G-code was accurate and correct. The experiment was performed on the obtained tool path based on ACO and the proposed method by Lin et al. [13]. At the end of the experiment, the cutting time was recorded for the purpose of machining time comparison as shown in Fig. 34b. Figure 35 shows the result of experiments.

The time of the roughing process obtained by the method proposed by Lin et al. [13] was 323 s. In our experiment, using the ACO and modified ACO, the times of roughing process were 297 and 294 s, respectively. Thus, it can be concluded that the modified ACO has reduced the cutting time during the roughing process by 7.2% compared with the previous method and 1.34% compared to the conventional ACO. The summary of the roughing time obtained based on optimisation and experiment is illustrated in Table 3. Overall, the roughing time based on the experiments is larger than the roughing time based on the optimisation process for all three methods. This is because, in the optimization process, the movement of the cutting tool between the two uncut regions is determined based on the *x*-axis and *y*-axis only. Meanwhile, in the actual machining process, the movement in the *z*-axis also involved a non-productive tool path.

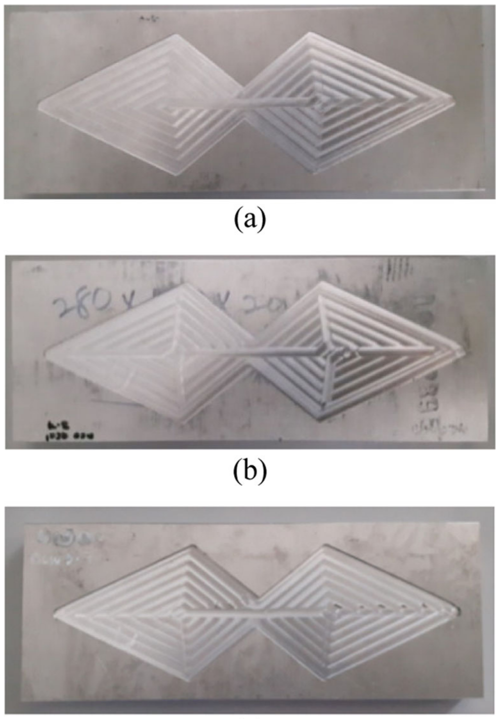


Fig. 35 Machining experiment. **a** Lin et al. (2013). **b** ACO. **c** Modified ACO

Table 3 Summary of roughing time

Method	T_m (optimization) (s)	T_m (Experiment) (s)	Differences (%)
Modified ACO	291	294	1.34
Conventional ACO	293	297	1.34
Lin et al. (2013)	312	323	3.2

8 Conclusion

One of the ways to increase the efficiency of contour parallel machining is by defining the tool path interval that is larger than the radius of the cutting tool. In addition, it can reduce the tool path length and machining time. However, the uncut region will occur at the corner and the centre of the contour. The uncut region can be removed by producing an additional tool path known as the clear tool path. In this paper, a new method of optimisation is introduced to minimise the clear tool path in order to reduce the entire tool path length and machining time in a roughing process. This optimisation solution involves three steps: construction of contour parallel offset, detection of coordinate of the uncut region, and minimising the tool path length using ACO. The proposed clear tool path was implemented in Matlab R2012b and the tool path generated based on this optimisation has been verified in an actual cutting experiment. The results indicate that the proposed clear tool path can be adapted in an actual cutting environment and is able to improve the cutting efficiency by reducing the machining time in the roughing process.

Acknowledgements The authors are thankful to the Ministry of Higher Education (MOHE) Malaysia for supporting this study under grant FRGS/2/2014/TK01 /UKM/02/1.

References

1. Hatna A, Grieve R, Broomhead P (1998) Automatic CNC milling of pockets: geometric and technological issues. Comput Integr Manuf Syst 11(4):309–330
2. Kim BH, Choi BK (2002) Machining efficiency comparison direction-parallel tool path with contour-parallel tool path. CAD Computer Aided Design 34(2):89–95
3. Choi BK, Park SC (1999) A pair-wise offset algorithm for 2D point-sequence curve. Comput Aided Des 31(12):735–745
4. Park SC, Choi BK (2001) Uncut free pocketing tool-paths generation using pair-wise offset algorithm. Comput Aided Des 33(10): 739–746
5. Park SC, Chung YC, Choi BK (2003) Contour-parallel offset machining without tool-retractions. Comput Aided Des 35(9):841–849
6. Lambregts CAH, Delbressine FLM, Vries WAH, Wolf ACH (1996) An efficient automatic tool path generator for D free-form pockets. Comput Ind 29:151–157
7. Lai W, Faddis T, Sorem R (2000) Incremental algorithms for finding the offset distance and minimum passage width in a pocket machining toolpath using the Voronoi technique. J Mater Process Technol 100:30–35
8. Kim HC, Lee SG, Yang MY (2006) A new offset algorithm for closed 2D lines with islands. Int J Adv Manuf Technol 29:1169–1177
9. Lee CS, Phan TT, Kim DS (2009) 2D curve offset algorithm for pockets with islands using a vertex offset. Int J Precis Eng Manuf 10(2):127–135
10. Choi BK, Kim BH (1997) Die-cavity pocketing via cutting simulation. Comput Aided Des 29(12):837–846
11. Choy HS, Chan KW (2003) A corner-looping based tool path for pocket milling. Comput Aided Des 35(2):55–166
12. Mansor MSA, Hinduja S, Owodunni OO (2006) Voronoi diagram-based tool path compensations for removing uncut material in 2½D pocket machining. Comput Aided Des 38:194–209
13. Lin Z, Fu J, Shen H, Gan W (2013) Global uncut regions removal for efficient contour-parallel milling. Int J Adv Manuf Technol 68: 1241–1252
14. Kumar S, Gupta AK, Chandna P (2014) Minimization of non-productive time during 2.5D milling. World Academy of Science, Engineering and Technology: International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering 8(6):1147–1152
15. Gupta AK, Chandna P, Tandon P (2011) Hybrid genetic algorithm for minimizing non productive machining time during 2.5D milling. Int J Eng Sci Technol 3(1):183–190
16. Oysu C, Bingul Z (2009) Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining. Eng Appl Artif Intell 22(3):389–396
17. Tian Y, Jiang P (2007) Optimization of tool motion trajectories for pocket milling using a chaos ant colony algorithm. IEEE International Conference on Computer-Aided Design and Computer Graphics. 389–394
18. Kiani K, Sharifi M, Shakeri M (2013) Optimization of cutting trajectory to improve manufacturing time in computer numerical control machine using ant colony algorithm. J Eng Manuf 228(7):1–6
19. Abbas AT, Aly MF, Hamza K (2011) Optimum drilling path planning for a rectangular matrix of holes using ant colony optimisation. Int J Prod Res 49(19):5877–5891
20. Ghaiebi H, Solimanpur M (2007) An ant algorithm for optimization of hole-making operations. Computers& Industrial Engineering 52: 308–319
21. Liu X, Hong Y, Zhonghua N, Jianchang Q (2013) Process planning optimization of hole-making operations using ant colony algorithm. International Journal Advance Manufactruing Technology 69(1):1–9
22. Rodríguez NM, Ross OM, Sepúlveda R, Castillo O (2012) Tool path optimization for computer numerical control machines based on parallel ACO. Eng Lett 20(1):1–4
23. Ross OM, Rodriguez NM, Sepulveda R, Melin P (2012) Methodology to optimize manufacturing time for a CNC using a high performance implementation of ACO. Int J Adv Robot Syst 9(121):1–10
24. Abbas AT, Hamza K, Aly MF (2014) CNC machining path planning optimization for circular hole patterns via a hybrid ant colony optimization approach. Mechanical Engineering Research 4(2):16–29
25. Purian FK, Farokhi F, Nadooshan RS (2013) Comparing the performance of genetic algorithm and ant colony optimization algorithm for mobile robot path planning in the dynamic environments with different complexities. Journal of Academic and Applied Studies 3(2):29–44
26. Dorigo M, Maniezzo V, Colorni A (1996) Ant system : optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics 26(1):29–41
27. Luo X, Yu F, Zhang J (2006) Study of parametric relation in ant colony optimization approach to traveling salesman problem. Computational Intelligence and Bioinformatics: 22–32. Springer Berlin Heidelberg
28. Abdullah H, Ramli R, Wahab DA, Qudeiri JA (2016) Tool path generation of contour parallel based on ant colony optimization. Jurnal Teknologi 78:31–36
29. Kim HC (2010) Tool path generation for contour parallel milling with incomplete mesh model. Int J Adv Manuf Technol 48:443–454