10.1109/CADCG.2007.4407914

Optimization of Tool Motion Trajectories for Pocket Milling Using a Chaos Ant Colony Algorithm

Ying Tian and Pingyu Jiang
State Key Laboratory for Manufacturing Systems Engineering
School of Mechanical Engineering
Xi'an Jiaotong University, Xi'an 710049, China
pjiang@mail.xjtu.edu.cn

Abstract

In order to generate the shortest and appropriate tool motion trajectories and reducing the non-productive tool positioning time, the tool motion trajectories of pocket milling with contour-parallel should be optimized. A chaos ant colony algorithm (CACA) is presented to solve the optimization problem in this paper. In the CACA, ant colony optimization (ACO) is used to find out the local optimal solution of current position and chaos iterative is used to extend the searching spaces. To demonstrate the efficiency of the CACA, a numerical experiment is put forward. The results show that the algorithm could realize the optimization of tool motion trajectories effectively and rapidly in pocket milling with contour-parallel.

1. Introduction

One of the most occurring industrial milling tasks is pocket machining [1]. Due to the fact that a large number of mechanical parts are 'terrace-shaped' or can be approximated by that, a 2.5D type of control is often used. We define a 2.5D free-form pocket as a planar area bounded by a closed outer contour that is design-extruded via the normal of that plane, resulting in a volume with constant depth. This volume may contain islands of unremoved material, defined by the inner contours [1]. The majority of industrial milling tasks can be performed using 2.5D pocket milling. More than 80% of all mechanical parts can be cut by applying the concept of pocket machining [2]. Hence, many researchers focused their attention on the generation algorithm of tool motion trajectories.

Tool paths can be classified into two major types: direction-parallel and contour-parallel. Contour parallel tool path is a coherent tool path in the sense that the cutter is kept in contact with the cutting material most of the time. So it incurs less idle time such as those spent in lifting, positioning and plunging the cutter [3]. In this paper, the method for optimizing the tool motion trajectories with contour-parallel is discussed.

In pocket milling with contour-parallel, in order to reduce the non-productive tool positioning time, machining sequence and tool motion trajectories should be optimized. To realize optimization of tool motion trajectories to generate the shortest and appropriate tool motion trajectories and reduce the non-productive tool positioning time, a chaos ant colony algorithm (CACA) is presented in this paper. An example demonstrates that the algorithm could realize the optimization of tool motion trajectories effectively and rapidly in pocket milling with contour-parallel.

2. Optimization problem of tool motion trajectories

In pocket milling with contour-parallel, optimization of tool motion trajectories is to find the shortest and the most appropriate tool motion paths from start-cutting position to end-cutting position. It involves the determination of the start-cutting position, machining sequence, lifting, positioning and plunging the cutter. The increase of the machining effectiveness can be achieved through an effective reduction of the tool path length. The improvement is usually the result of the corresponding reduction in the total machining time.

Contour profiles can be formed by "pair-wise intersection" or "Voronoi diagram" [3]. Productive machining paths are determined by contour profiles. The contour profiles of a pocket are shown in figure 1, in which $v_1, v_2, \cdots, v_{11}$ denote the machining loops. Each machining loop is a successive cutting trajectory. These machining loops are productive machining paths.

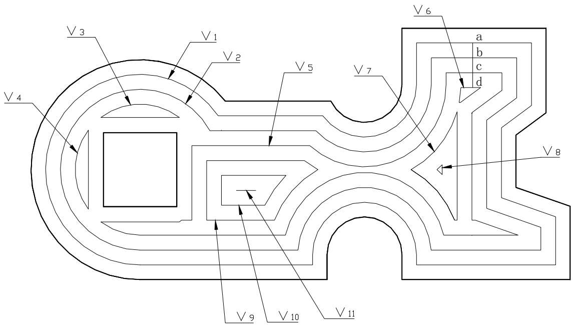

Figure 1. Contour profiles of a pocket

The machining tree is generated along with generated contour profiles. The machining tree is shown in figure 2. Each node of the machining tree is mapped by a machining loop in the contour profiles.

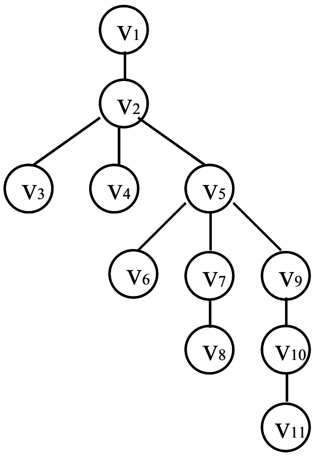

Figure 2. Machining tree

Some additional traversing motions are necessary for overstepping the gaps between successive cutting trajectories. These additional traversing motion paths are non-productive machining paths in cutting processes. At the end of each machining loop, a lead-segment is necessary to lead cutter to enter next machining loop. The lead-segment is a non-productive machining path. It may increase non-productive tool positioning time if machining sequence is not appropriate. In addition, it is very important to select start-cutting position. It may avoid more non-productive paths while more proper start-cutting position is selected.

When cutter is about to enter an appointed machining loop at the end of a branch of the machining tree, such as v_3, v_6, v_8 or v_{11}, a tool motion route is determinate. Here, the tool motion route is the path where the distance is shortest from current position of cutter to the appointed machining loop. Also the cutting position which cutter would enter is determinate. For example, in the pocket shown in figure 1, suppose that start-cutting position is *a*, initial machining sequence is 'v_1-v_2-v_5-v_6'. The tool motion trajectories is shown as "*a*-v_1-*ab*-v_2-*bc*-v_5-*cd*-v_6-*d*". Then the candidates for next machining loop are v_3, v_4, v_7 and v_9 when cutter is on position *d* after v_6 has been machined. The facts that which loop would be selected as the next machining loop are relative to the current position *d* of cutter.

The minimum distances from current position to candidates can be expressed as formula (1).

$$Dist = d_{\min vi} \tag{1}$$

where $d_{\min vi}$ is the minimum distance from current position to candidate v_i.

The machining loop corresponding to the minimal value of *Dist* will be selected as the next machining loop. The objective function is expressed as equation (2).

$$F = \min \sum Dist \tag{2}$$

The objective function is to minimize the summation of $Dist$. Our research is to find out appropriate start-cutting position and machining sequence to make the non-productive path be minimum.

3. Optimization of tool notion trajectories for pocket milling using a chaos ant colony algorithm

A meta-heuristic algorithm on the base of ants' behavior was developed in early 1990s by Dorigo, Maniezzo, and Colorni. They called it ant colony optimization (ACO) because it was motivated by ants' social behavior. Ants are capable of finding the shortest path from food source to their nest or vice versa by smelling pheromones which are chemical substances they leave on the ground while walking [4].

The chaos optimization algorithm (COA) is a new kind of searching method. The basic idea of the algorithm is to transform the variable of problems from the solution space to chaos space and then perform search to find out the solution by virtue of the randomicity, orderliness and ergodicity of the chaos variable [5].

In this paper, we present a novel CACA to solving the optimization problem on the basis of integrating the respective characteristic of the COA and ACO. ACO, the basic of the algorithm, is used to find out the local optimal solution of current position. Chaos iterative is used to generate more positions. In other words, chaos iterative is to extend searching spaces. The use of chaotic sequences is more capable of escaping from local optima than stochastic ergodic searches that depend on probabilities.

Initial machining loop are two-dimensional freedom close-curve, so the number of start-cutting position is infinitude in theory. The solving processes for the optimization problem can be described as follows by CACA:

Step 1:Divide initial machining loop into p sub-areas averagely. Generate an initial position randomly on each sub-area. Put m ants on each generated position as a work ant colony.

Step 2:Apply ACO to each work ant colony to obtain local optimal solution of each position. This may be regarded as the local optimum of this sub-area.

Step 3: Generate new searching space of each sub-area by chaos iterative.

Step 4:Use ACO to search optimal solutions further in new searching spaces according to different probability p_i.

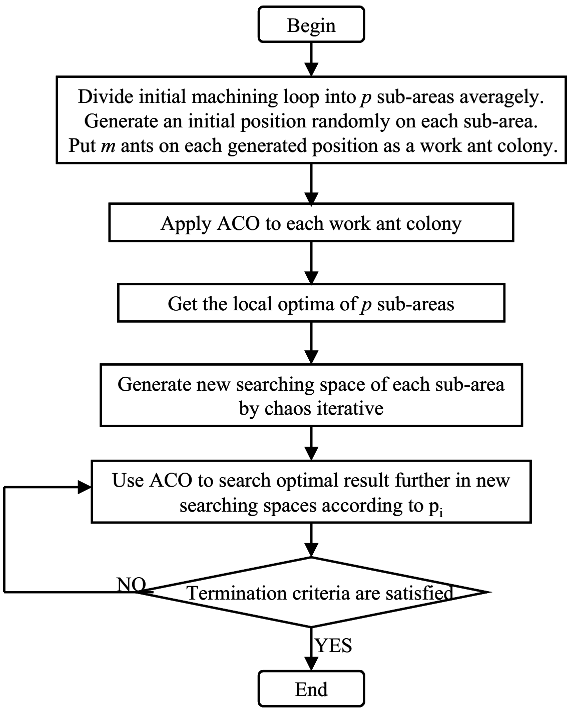

Figure 3. The flowchart of the CACA for the optimization of tool motion trajectories

Step 5: If the termination criteria are satisfied, then stop. Otherwise, go to Step 4.

The flowchart of the CACA for the optimization of tool motion trajectories is shown in figure 3.

The keys of proposed CACA are ACO and chaos iterative, which are introduced below.

3.1. Ant colony optimization for finding out the local optima

ACO is for finding out the local optima of the current positions. The first m ants are put on each initial position as a work colony. Each of the ants transfers to next machining loop according to state transition rules. Machining loops, which have been passed through, are put into tabu tables. Machining loops existing in tabu tables cannot be regarded as candidates later. Transition probability is in proportion to visibility and the concentration of pheromones. Nearer the path is, greater the visibility is. The paths with greater visibility are easier to be selected as next machining loop. Also, the paths with larger concentration of pheromones are easier to be selected. The paths that all ants pass through must be same after repetition time after time. This is the local optimum for that initial position.

The state transition and pheromone updating rules are used in the ACO. These rules are determined by the following approaches. The remainder of this section will describe the algorithm in detail.

3.1.1. Transition rules. Suppose i is the current position of the ant k. In accordance with the facts, if the machining loop with i has one or several sub-loops, the ant k will transfer unconditionally to a sub-loop to which the distance is shortest from i.

If the machining loop with i has no any sub-loops, the ant k will transfer to an un-machined loop according to transition probability. Transition probability can be formulated as follows.

$$p_{ij}^{k} = \begin{cases} k_1 \dfrac{1}{d_{ij}} + k_2 \dfrac{\tau_{ij}}{\sum\limits_{s \in allowed_k} \tau_{is}} & j \in allowed_k \\ 0 & otherwise \end{cases} \tag{3}$$

where d_{ij} is the shortest distance from i to the candidates for next machining loop, on which j is the nearest position to i.

$\dfrac{1}{d_{ij}}$ is the visibility between j and i.

k_1 is the visibility coefficient. k_2 is the pheromone coefficient. The visibility has important effects on transition process, so we let $k_1 \geq 4k_2$.

allowed$_k$={the sub-loops whose father-loops have been passed through}- tabu$_k$, is the candidates for next machining loop.

τ_{is} is the amount of pheromones on path (i, j).

3.1.2. Pheromone updating rules. The updating of pheromones can be performed by three ways: after each move, after completion of solution, or at the end of each iteration [6]. The third way is adopted in this paper.

Initially, the values of pheromones on all the paths are set to be equal. Pheromone updating is performed after *m* ants complete each iteration.

$$\tau_{ij}^{t} = \tau_{ij}^{t-1} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{4}$$

where $\Delta\tau_{ij}^{k}$ is the amount of pheromones left on path （*i,j*） by ant *k* at the end of iteration *t*.

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{C}{L_k} & \text{If ant k pass through path } (i, j) \\ 0 & \text{else} \end{cases} \tag{5}$$

Where *C* is a constant. L_k is the length summation of all paths passed through by ant *k* in iteration *t*.

3.1.3. Termination condition. The iteration is repeated until some termination conditions are met, such as a maximum number of iterations has been performed or a satisfied solution is found. The algorithm stops as the optimal solution is found, or 2000 iterations are performed.

3.2. Chaos iterative for extending searching spaces

Extending searching spaces should be performed after local optima are obtained. Chaotic sequences are generated according to Logistic chaos iterative mapping expressed as formula (6). The chaotic sequences compose the new searching spaces of every sub-area.

$$z(n+1) = \mu \cdot z(n)(1.0 - z(n)) \tag{6}$$

where z is the chaos variable, $0 \leq z(n) \leq 1$; n is the times of iteration, *k*=1,2,⋯; μ is the control parameter. It is easy to testify that the system is entirely in chaos situation when $\mu = 4$ and the chaos space is [0,1].

ACO is used to search optimal solution further in new searching spaces according to different probability. The probability of a sub-area chosen to search further is expressed as formula (7).

$$p_i = \frac{\left|\dfrac{1}{f_i^*}\right|^{\alpha}}{\sum_{s=1}^{p} \left|\dfrac{1}{f_s^*}\right|^{\alpha}} \tag{7}$$

where f_i^* is the local optimum of sub-area i. α is the relative importance of each local optimum. Apparently, the smaller the local optimum is, the larger the probability of the sub-area chosen to search further is.

3.3. Minimum distance from the current position to the next machining loop

The minimum distance from *k*-ant current position to the next machining loop, d_{ij} may be solved through the theory of differentiable geometry when the machining loop with *k*-ant has no any sub-loops.

Assume that the current position of *k*-ant is $M(x_0, y_0)$ as shown in figure 4. The next machining loop is a free-form curve, which expressed by the single variable *x*. This problem is to seek the minimum distance from M to the free-form curve.

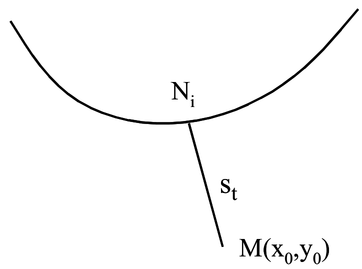

Figure 4. Minimum distance form M to the free-form curve

Given the searched point $N_i(x(t), y(t))$ is on the free-form curve at time t. s_t expresses the distance between M and N_i. The tangential vector through the point $N_i(x(t), y(t))$ is r_x at time t, then this curve's differential equation is show as formula (8):

$$dy = r_x dx \tag{8}$$

Formula (9) will be obtained through the distance formula between two points:

$$s_t^2 = [x(t) - x_0]^2 + [y(t) - y_0]^2 \tag{9}$$

Formula (10) will be obtained through differentiating formula (9):

$$(s_t^2)' = 2[x(t) - x_0]x'(t) + 2[y(t) - y_0]y'(t) \tag{10}$$

Formula (11) will be obtained through leting equation (10) equal to zero:

$$2[x(t) - x_0]x'(t) + 2[y(t) - y_0]y'(t) = 0 \tag{11}$$

Combining (8) with (11), we solve $x(t)$ and $y(t)$, then substitute them into (9) for obtaining s_t which is the minimum distance from M to the free-form curve at time *t*.

We take this step length $dx(t)$ and $dy(t)$ respectively, at time $t+1$, the point N_{i+1} is

$$\left.\begin{aligned} x(t+1) &= x(t) + dx(t) \\ y(t+1) &= y(t) + dy(t) \end{aligned}\right\} \tag{12}$$

Iterate formula (9) and calculate s_{t+1} until $(t+n)nd$ iterative step length is less than given error ε. This s_{t+n} is exactly the minimum distance that we expect and the point $(x(t+n), y(t+n))$ is the position which is the nearest position to M on the free-form curve.

4. Experiment and results

We use a numerical experiment to test the CACA for the optimization of tool motion trajectories for the pocket shown as figure 1. The part drawing of the pocket is shown as figure 5.

The optimal result of tool motion trajectories using CACA is shown in figure 6. The value of the objective function is 129.11. The paths of lifting, positioning and plunging the cutter are expressed as dashed.

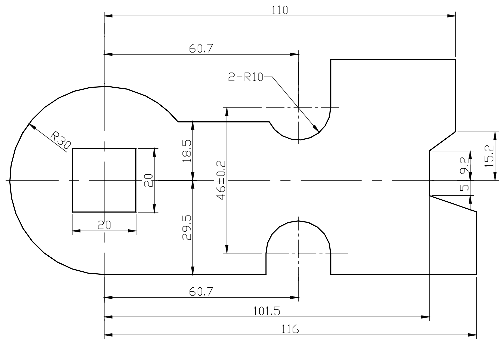

Figure 5. Part drawing of the pocket

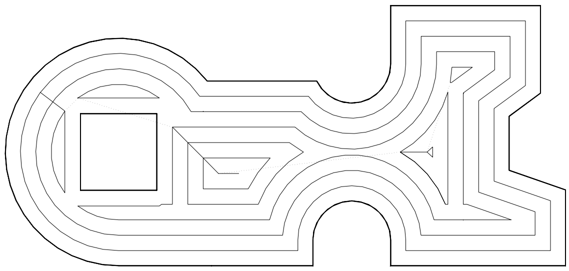

Figure 6. Optimal result of tool motion trajectories using CACA

5. Conclusions

In order to realize optimization of tool motion trajectories to reduce the non-productive tool positioning time in pocket milling with contour-parallel, a chaos ant colony algorithm (CACA) is presented in this paper. In the proposed algorithm, ACO is used to find out the local solution of current position and chaos iterative is used to extend searching spaces. The main advantage of chaotic sequence is that it can extend the diversity of the searching spaces. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastically ergodic searches that depend on probabilities. The use of chaotic sequences can be helpful to escape more easily from local optima. To demonstrate the efficiency of the CACA, a numerical experiment is put forward. The results indicate that the CACA could realize the optimization of tool motion trajectories effectively and rapidly in pocket milling.

Acknowledgments. This project is under the support of both "National 863 High-Tech R & D Program (Project No.: 2006AA04Z149) and the Program for New Century Excellent Talents in University by China

Ministry of Education (Project No.: NCET-04-0928)”.
We hereby thank them for the financial supports.

References

[1] C.A.H. Lambregts, F.L.M. Delbressine, W.A.H. de Vries, A.C.H. van der Wolf, “An efficient automatic tool path generator for 2 1/2D free-form pockets”, *Computers in Industry*, Elsevier Science B.V., Amsterdam, Netherlands, 3 (1996), pp. 151-157.
[2] Y.S. Lee, T.C. Chang, “Application of computational geometry in optimizing 2.5D and 3D NC surface machining”, *Computers in Industry*, Elsevier Science B.V., Amsterdam, Netherlands, 1 (1995), pp. 41-59.
[3] H.S. Choy, K.W. Chan, “A corner-looping based tool path for pocket milling”, *Computer-Aided Design*, Elsevier Science Ltd, 2 (2003), pp. 155-166.
[4] K. Eshghi, M. Kazemi, “Ant colony algorithm for the shortest loop design problem”, *Computer and Industrial Engineering*, Elsevier Ltd, Oxford, United Kingdom, 4 (2006), pp. 358-366.
[5] Z. Guo, S. Wang, J. Zhuang. “A novel immune evolutionary algorithm incorporating chaos optimization”, *Pattern Recognition Letters*, Elsevier, Amsterdam, Netherlands, 1 (2006), pp. 2-8.
[6] M. Solimanper, P. Vrat, R. Shankar. “Ant colony optimization to the inter-cell layout problem in cellular manufacturing”, *European Journal of Operational Research*, Elsevier, 3 (2004), pp. 592-606.