Segmentation of Cutting Path Based on Feed Rate Interval

XIEGUO XIE, GUANGRONG YAN, XIANGYU XU and ZAIJUN WANG

ABSTRACT

This paper provides an optimization method of cutting path segmentation. For given CNC machining codes, the sampling point sequence of cutting path is processed and optimized by using the greedy method or the dynamic programming method in the case of basically satisfying the practical requirement. Reasonable and effective feed rate intervals can be obtained by the optimization.

INTRODUCTION

At present, the NC processing codes, run by CNC machine to complete the parts processing, are mostly written by graphical interactive software, such as Master CAMCimitron, UG, Power Mill, etc. Nevertheless, these softwares have some limitations in optimization of the processing parameters of individual parts. The spindle speed of the machine, feed speed and machining depth are determined by the programmer, which is affected by the experience and professional level of the programmer. The programmer usually choose a more conservative feed rate based on the material of the machining workpiece and the size and material of the tool. As the milling force and other physical changes in processing are ill-considered, the machine cannot display the best performance.

In the milling process, the feed rate and spindle speed at each sampling point are constant for the same sequence on the cutting track corresponding to a piece of G code. But the cutting depth and cutting width at each sampling point may gradually change with the progress of the processing. Some physical quantities such as the cutting force and cutting temperature at each sampling point, would be unable to meet the required constraints at the same time. Therefore, the feed rate used by the G code may not be reasonable for all cutting positions along the cutting path.
PROBLEM ANALYSIS

In the milling process, when the feed rate used by the G code is not reasonable for all cutting positions in the cutting tracks, it is necessary to segment the cutting track corresponding to the G code. After each segment of the cutting, the feed rate used by the path allows each cutting track to meet the physical constraints. The segmentation principle is shown in Figure 1.

![Figure 1. Segmentation of cutting track.](image)

To segment the cutting track, it is necessary to calculate interval of the reasonable feed rate of each sampling point firstly and then use an appropriate algorithm to divide the sampling point sequence into N segments. For each sampling point sequence segmented, feed rate intervals of all sampling point of the sequence in the segment is exactly the reasonable feed rate interval of the sampling point sequence, which cannot be an empty set. In the process of segmentation, the smaller N value is, the less G code statement is optimized and the better effect of segmentation can be obtained. If G code is generated point by point instead of segmenting the cutting path, the efficiency of the optimization process will be greatly reduced, resulting in a too large G code file.

To sum up, it can be seen that this is a classic problem of cluster analysis, however, the constraints of which is different from the traditional cluster analysis. First, points of cluster analysis must be continuous; second, the reasonable interval of feed rate of the cluster sampling point cannot be empty. In general, the greedy method and the method of dynamic programming can be used to solve the complicated clustering problem. The greedy algorithm can guarantee the relatively good times, but only get sub-optimal solution. However, suboptimal solution usually can meet the needs of engineering practice. The method of dynamic programming is more complex than the greedy one and takes a longer time, yet it guarantees the optimal solution. If the optimal solution is needed in the engineering application, this method can be adopted.

METHOD OF GREEDY SEGMENTATION

Greedy algorithm usually starts from the local information, regardless of the global optimality of the solution, and do the best choice in the current. On the issue of the sampling point for clustering, the greedy algorithm starts from the first sampling point, step-by-step aggregation of polymerization points. When the new point cannot be
aggregated, the point that has been aggregated is treated as a segment. The next new point continues as a starting point for the new segment until all points are processed. The specific algorithm is described as the following.

Suppose that the sampling points of the current cutting path contain \( n \) sampling points. Each sample point is represented by the symbol of \( P_i \), and the reasonable interval of feed rate of \( P_i \) is denoted by \((V_{fimin}, V_{fimax})\). At the sampling point, the reasonable range of feed rate is

\[
V_{fimin} \leq V \leq V_{fimax},
\]

where \( i = 1, 2, 3, ..., n \). The process of greedy segmentation is as follows.

Step 1: Set the point number \( i = 1 \) of current processing, the \( M \) of current sampling point sequence is empty set, \( L = (-\infty, +\infty) \) (\( L \) represents the intersection of the reasonable interval of feed rate \( l \) of all sampling points in \( M \) set).

Step 2: Calculate the intersection of the reasonable interval \((V_{fimin}, V_{fimax})\) of the \( P_i \) sampling point and \( L \), marked as \( R \).

Step 3: If \( R \) is an empty set, the sampling point sequence in set \( M \) cannot be compatible with \( P_i \). All the sampling points in output \( M \) are taken as an aggregate segment, then \( M = \{ \} \), \( L = (-\infty, +\infty) \) and jump to step 5.

Step 4: If \( R \) is not an empty set, the sampling point in the set \( M \) can be compatible with \( P_i \). Add the point \( P_i \) to the set \( M \), set \( L = R \), \( i = i + 1 \) and jump to step 5.

Step 5: If \( i \) is equal to \( n + 1 \), it means that all the sampling points have been processed. The sampling points in \( M \) are output as an aggregate segment, and the whole process finishes.

Step 6: If \( i \) is less than or equal to \( n \), there are still sampling points to deal with, then jump to step 2.

The above process is the method of greedy segmentation. In general, the process of segmentation starts from the first point, followed by the sampling point that can be fused. When there is no new sampling points can be merged, those fused points are regarded together as a section. The sampling points not integrated would be considered as the beginning of new section, continuing to integrate later until all the sampling points are processed. The above process can be represented by a program flow chart as shown in Fig. 2.

As shown in Fig. 3, there is a reasonable interval of feed rate at each sampling point on a cutting path with 10 sampling points (the vertical line on each sampling point in the horizontal axis represents a reasonable interval of feed rate). From the beginning of the serial number of 1 sampling points to the serial number of 7, as the reasonable interval of feed rate of the first 7 sampling points would become empty set, the serial number from 1 to 6 sampling points are considered as one section. With the serial number of 7 as a new start, continue the integration until all the sampling points are fused. Finally, the sampling sequence can be divided into two segments, namely \( \{1, 2, 3, 4, 5, 6\} \) and \( \{7, 8, 9, 10\} \). On the whole, the greedy segmentation is simple and its complexity is \( O(n) \), while cannot secure an optimal solution. For some cases where the optimal solution is necessary, the method of dynamic programming segmentation will be put to good use.
Start

Enter the number n of sampling points

Enter a reasonable feed rate ranges for all sampling points

Initialize the number i = 1 of current sampling point

Initialize the set M = {} of current sample point sequence

Initialize the reasonable interval L = ( , ) of feed rate for the current sample point sequence

whether R is an empty set

Outputs all sample points in set M as an aggregate segment

Update M = {} L = ( , )

Whether i = n + 1

Outputs all sample points in set M as an aggregate segment

Add the sampling point numbered i to the set M

Update L=R

i=i+1

End

Figure 2. Chart of greedy segmentation flow.
Although the method of greedy segmentation can deal with the problem of sampling point segmentation with high efficiency, it cannot guarantee the optimal solution. Then the method of dynamic programming, often used to solve the problem with optimality, would be a fine choice. In the clustering problem of sampling point sequence, the minimum number of segments is the optimality property required in this problem. Thus this problem can be solved by the method of dynamic programming. The essence of method of the dynamic programming is to use the idea of Divide & Conquer, which break down the big problem into smaller similar sub-problems. When all the sub-problems are recursively solved, the solution of the sub-problem is combined into that of original problem. The solution of the repeated sub-problems is preserved in the process of solving, which can greatly reduce the complexity of the problem.

By analyzing the segmentation problem of the sampling point, the sub-problem of the problem is the strategy of optimal segmentation for solving all the sequence points from the i-th sequence point to the last sequence point. The specific algorithm is described as follows.

The sampling point sequence of the current cutting track contains n sampling points. Each sample point is represented by the symbol $P_i$, $i = 1, 2, 3, ..., n$, $P_i \in (V_{f_i, \min}, V_{f_i, \max})$. DP $[i]$ is the optimal strategy of segmentation from the i-th point to the n-th point of the solution, described in the form of a set. For example, the set of optimal segments of the point sequence from the first point to the fifth point can be expressed as $\{(1, 3), (4, 5)\}$, representing strategy for the 1, 2, 3 points of a segmentation and the 4, 5 points for another segmentation, including $|DP [i]|$ as the number of optimal segmentation strategy. The specific procedure of the algorithm are as follows.

Step1: Initialize the sequence of the current sampling points, $i = n - 1$. Initialize DP $[n]$ to $\{(n, n)\}$.
Step2: If $i$ is less than or equal to 0, end the program; otherwise initialize the sequence number $j = i$ and initialize $\{DP [i] = \{(i, i), (i + 1, i + 1), ..., (n, n)\}\}(The worst case, only one sampling point divided into one section).
Step3: If $j$ is greater than $n$, jump to Step 8; otherwise calculate the intersection $R$ of the feed rate range from the sequence of sampling point from $i$ to $j$.
Step4: If $R$ is empty, assign $i = i - 1$. Go to step Step2.
Step5: If $R$ is not empty, calculate $\text{Cmp} = |DP [i]| - (1 + |DP [j]|)$.
Step6: If Cmp is greater than 0, update $DP [i] = \{(i, j)\} \cup P [j + 1]$.  

Figure 3. Example of greedy segmentation.
Step7: Update \( j = j + 1 \), then jump to Step3.
Step8: Update \( i = i - 1 \), then jump to Step2.

After the above program, DP \([1]\) will save the optimal segment strategy of sequence of sampling points. The chart of algorithm flow is shown in Fig. 4.
The algorithm complexity of the method of dynamic programming segmentation is analyzed and the result is that each DP [i] is \(O(n - i)\). The DP [1] is required to obtain all DP [i], \(2 \leq i \leq n\), then the total complexity of the algorithm is \(O(n \times n)\). It can be seen that this algorithm runs much faster than that of greedy segmentation.

CONCLUSION

This paper introduces the technique of cutting path segmentation based on the reasonable feed rate range of sampling points in the optimization system of milling process code. Through the analysis of the problem, two methods of greedy segmentation and dynamic programming segmentation are proposed and the operation efficiency of these algorithms are particularly analyzed. Method of greedy segmentation features fast operation speed while cannot guarantee the optimal solution, basically meeting the practical needs of the project. Method of dynamic programming segmentation, with a slower operation speed, has the optimal solution and is suitable for the occasions of more optimization requirements.

ACKNOWLEDGEMENTS

The research work was financially supported by the Project named Technology and Equipment of NC Machining Unit for Aircraft Complex Structure (No. 2013 ZX04001031).

REFERENCES

1. Junxiang Wang. Research on the principle and application of dynamic programming algorithm [J]. Computer Knowledge and Technology, 2007, (36): 150-151.
2. Rui Li, Fucai Qian, Li Li, Jianjun Gao. Research on dynamic programming problem [J]. Systems Engineering Theory and Practice, 2007, (8): 56-64.
3. Honglin Guo. The relationship between the optimality theorem and the optimization principle of dynamic programming and its inference [J]. Public Science (Scientific Research and Practice), 2007, (18): 28.
4. Coleman, Jinui Pan. Introduction to the algorithm [M]. Machinery Industry Press, 2006.
5. Yang Li, Yongcheng Zhao, Lan Wei. Overview of NC machining simulation technology [J]. System Simulation Technology, 2008, 4 (2): 111-116.
6. Yongzhi Pan, Xing Ai, Zhita Tang, et al. Optimization of Tool Geometry Parameters and Cutting Parameters Based on Cutting Force Prediction Model [J]. China Mechanical Engineering, 2008, 19 (4): 428-431.