Empty tracks optimization based on Z-Map model

Le Liu a, Guangrong Yan b, Zaijun Wang, Genao Zang

School of mechanical engineering and automation, Beihang University, Beijing 100191, China

a liule_123@163.com, b yangr@buaa.edu.cn

Abstract. For parts with many features, there are more empty tracks during machining. If these tracks are not optimized, the machining efficiency will be seriously affected. In this paper, the characteristics of the empty tracks are studied in detail. Combining with the existing optimization algorithm, a new tracks optimization method based on Z-Map model is proposed. In this method, the tool tracks are divided into the unit processing section, and then the Z-Map model simulation technique is used to analyze the order constraint between the unit segments. The empty stroke optimization problem is transformed into the TSP with sequential constraints, and then through the genetic algorithm solves the established TSP problem. This kind of optimization method can not only optimize the simple structural parts, but also optimize the complex structural parts, so as to effectively plan the empty tracks and greatly improve the processing efficiency.

1. Introduction

In the milling process, the NC code generated by CAD / CAM software still has the problem that the cutting depth is too large or too small, the feed speed is not reasonable enough, so the need to optimize the generated NC code. For the optimization of NC code, Shandong University Pan Yongzhi [1] and others use the genetic algorithm to optimize the geometrical parameters and cutting parameters of the tool used in the process. Liu Haijiang and Huang Wei [2] of Tongji University passed the particle swarm intelligence algorithm, The optimization of cutting parameters is realized. Huang Zhengtao, State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology [3] and others, use GSA algorithm (gravitational search algorithm) to optimize the cutting parameters.

The above research is mainly aimed at the optimization of cutting parameters, but the path optimization of the empty stroke is not carried out. And for the inclusion of many features of the parts, the processing process includes emptier trajectory. These path, if not well planned, Will cause the empty stroke path is too long, reducing processing efficiency.

There is a class of optimization algorithms specifically designed for porous machining of empty travels. This typical optimization problem is a typical traveling salesman problem (TSP). Traveler problem is a classic combinatorial optimization problem, but this optimization method cannot be used to optimize the trajectory optimization for a wider range of milling processes. The optimization method developed in this paper is not only to optimize the trajectory of the empty part of the simple parts, but also to optimize the empty trajectory of complex structures.
The Z-Map model is a geometric model of a part that describes the amount of components, which consists of a large number of discrete Z-Map points \(^{[4-5]}\). After a detailed analysis of the nature of the empty stroke optimization, this paper proposes an empty stroke optimization technique based on the Z-Map model technique. The problem of the empty stroke optimization is transformed into a TSP with sequential constraints, which can be applied to a wide range of empty stroke optimization. The rough process of the algorithm is as follows:

1. Using the sampling point information of the cutting track, the Z coordinate value of the safety plane used for the cutting track is obtained.
2. The cutting path is divided into a lot of unit processing section in accordance with the principle of falling from the safety plane - feed processing - lift the knife back to the safety plane.
3. Using the Z-Map model of cutting simulation technology, get the successive constraint information between each unit processing section.
4. Using the genetic algorithm to re-plan the processing sequence of the unit processing section, so that the re-planning unit between the empty section of the flight path can be shortest, and the unit processing segment should be satisfied the sequential constraints.

In summary, the Z-Map model based on the trajectory planning technology flow chart shown in Figure 1.

![Figure 1](attachment:Figure_1.png)

**Figure 1** Vectorization of Trajectory Planning Based on Z-Map Model

2. Access to Safety Plane
The Z coordinate value which determines the safety plane of a cutting track can be judged by the statistical data of the sampling point information on the cutting track. The specific treatment is as follows: traversing all the sampling points on the cutting track, all the cutting depth and cutting width whose value is zero and Z coordinate values of the same sampling points attributed to a class, ignoring the sampling points whose cutting depth and cutting width is not 0. The Z coordinate value of the safety plane is the Z coordinate value corresponding to the largest number of sampling points.

The reason why this method can be used to determine the location of the safety plane is that in the well-generated cutting trajectory, the safety plane will have a lot of linear empty stroke trajectory, and in the case of sampling distance of 1mm, these traces are separated into a number of sampling points with a cutting depth and a cutting width whose value is zero, and the Z coordinate values of these sampling points are consistent with the Z coordinate values of the safety plane. Of course, there are
some special cutting trajectories may not meet the above situation, in these cases, you can manually specify the Z-coordinate value of the security plane, and then process follow-up optimization.

3. Marking of The Unit Processing Section

After obtaining the Z coordinate values of the safety plane, the cutting trajectories can be divided into a section of the unit processing section by traversing the sequence of sample points. As shown in Figure 2, the unit machining section refers to a complete machining path from the safety plane to the feed cutting and then lift tool back to the safety plane.

![Figure 2: Segment of the Unit Processing Section](image)

Each segment of the unit processing section, with $AtomLine_i$ to represent, where $i$ is the serial number of the serial number. The process of finding the unit processing section is as follows: i. find the first the sampling point as the starting point of the current unit processing section; ii. follow the sampling points are added to the unit processing section, until there is a sampling point of the Z coordinate value back to the safety plane, set the sampling point as the end of the unit processing section, thus a complete unit processing section was found; iii. repeat the above process. When traversal all the sampling points, all the machining sections of the cutting track are found.

4. The Order Constraint Between Unit Processing Segments

Using the above method, all the found machined sections will cut the blank model, but different unit sections may cut the same area of the blank model, unless the order of cutting is different. If this is the case, there will be the order between the machined sections for cutting the same area of the blank. In the case of optimization, the post-machined unit machining section cannot be machined in the machined section before cutting, otherwise it may cause serious processing problems. The solution to this constraint can be achieved by using the cutting technique based on the Z-Map model.

In the Z-Map model technique, the cutting of the blank model is the process of updating the Z coordinate values at the time of cutting the tool at each Z-Map grid point.

If there are N-segment unit processing sections, all the values of the two-dimensional array which is represented by map $[N][N]$ are 0. If map $[i][j] = 1$ ($1 \leq i \leq N, 1 \leq j \leq N$), it indicates that the unit segment of the i-th unit must precede the j-th segment of the unit. We use the graph model to represent the relationship between the processing segments of each segment. The method of filling the two-dimensional array which is represented by map $[N][N]$ is achieved by cutting the machining of the blank Z-Map model by the processing section of each unit in the order of the original cutting track. The process is as follows:

1. Set the current processing section for the i-th unit.
2. Using the geometric simulation technique of the Z-Map model, let the tool cuts along the Z-Map model of the blank along the i-th unit machining section, and saves all the Z-Map grid points whose the Z coordinate values are updated as a set of M.
3. Traverse all Z-Map grid points in set M and do the following for each Z-Map grid:
Step1: Get the serial number of the upper segment processing section recorded at the Z-Map grid point, denoted as j.
Step2: If j is not 0, update map \([j][i] = 1\).
Step3: update the Z-Map grid point record of the upper unit processing section number i.
When all the unit segments are processed in accordance with the above process, the map records the contingency relationships that should be met between all the unit segments.

5. Optimization of TSP Problem with Sequential Constraints Based on Genetic Algorithm
After using the above-mentioned method to obtain the successive constraint relation between the processing sections of each unit, the problem of how to sort the successive processing order between the processing sections of each unit so that the empty stroke cutting path in the safety plane is the shortest can transfer into the order-constrained TSP problem. For the solution of the TSP problem, we can use the genetic algorithm\(^{[6-8]}\) to solve it.

Genetic algorithm is a kind of intelligent search algorithm which simulates the evolutionary environment of natural evolution. The basic framework of genetic algorithm needs to realize gene coding and decoding, fitness function, cross mutation operation and selection operation. In the following section, the detailed introduction of the process of solving the TSP problem with sequential constraints by using genetic algorithm is shown. Using genetic algorithm to solve the problem of the general process block diagram is shown in Figure 3.

**Figure.3** General Flow Diagram of Genetic Algorithm

5.1. Gene Coding and Decoding and Initialization
In the genetic algorithm, the coding of the gene is used to represent the solution that needs to be optimized. In the problem of the optimization of the trajectory empty stroke discussed in this paper, if the number of unit segments included in the cutting trajectory is N, the gene coding can be encoded by a N dimensional vector to represent. For example, an empty stroke optimization problem with five unit segments can be expressed by \((0.1, 0.2, 0.3, 0.35, 0.05)\) in the solution space. In order to make the real vector of each dimension contained in the vector do not diverge, and let the crossover operation defined later have more practical significance, set a constraint on the gene vector, that is, the real number of each dimension must be greater than 0, and the sum of all the real numbers of the dimension is 1. How to obtain the processing sequence of the processing section represented by the gene vector \((0.1, 0.2, 0.3, 0.35, 0.05)\) is the process of gene decoding, as described in detail in Table 1.
From the information shown in Table 1, it can be seen that each dimension of the gene vector represents a unit processing segment. The size of each dimension's real number represents the processing order of the unit processing section represented by the dimension in all unit processing segments. The smaller the number is processed first. Therefore, the process of gene vector decoding is to sort all the dimensions of the gene vector in the order from small to large, and then the processing segments represented by each dimension are processed in the order of sorting.

In the process of initializing the population, it is necessary to randomly generate the available gene vectors using the random function. This can be carried out by the following procedure, using the random number of random numbers between 0 and 1. Each dimension of the gene vector is filled with random numbers, but this cannot make all the sum of the dimensions of the gene vector is 1, so it also is necessary to divide all dimensions by the sum of all the dimensions of the original gene vector after filling, so that the normalization can be done. After the above operation, the randomly generated gene vector satisfies the constraint that the sum of all the dimensions is 1, but the processing order of the unit processing segment represented by the decoding may not satisfy the order constraint between the unit processing segments that need to be satisfied. In order to make the gene vector satisfy the order constraint between the unit segments, it is necessary to use the relaxation method to correct the values of the different dimensions of the gene vector. The process is to traverse the two sections of the unit processing segment between the constraints. If the order of the dimensions of the two-unit processing segments in the original gene vector is opposite to the constraint, then the real values of the two dimensions are exchanged. Repeat the above check until the processing sequence represented by the gene vector satisfies all the unit processing segments. The following is a simple example shown in Figure 4 illustrates how to use the relaxation method to modify the vector.

5.2. Fitness Function
For each gene vector Gene, how to determine the merits of the individual gene, needs to have a measurement method. In fact, the design of the fitness function is easy, directly using the corresponding solution, calculate the length of the empty stroke trajectory on the solution plane, the smaller the length of the empty trajectory, indicating that the gene is better, so the empty stroke trajectory length of the
reciprocal to represent the merits of gene can be. The greater the fitness function, the better the gene. The specific calculation method is as follows:

\[ f(Gene) = \frac{1}{\sum_{i=1}^{N-1} |\text{AtomLine}_i\text{AtomLine}_{i+1}|} \]

In the formula, \( N \) is the number of the unit processing segments. \( \text{AtomLine}_i \) is the unit of the \( i \)-th unit of the corresponding solution, referring to the straight line distance from beginning of the \( \text{AtomLine}_i \) to the beginning of the \( \text{AtomLine}_{i+1} \).

5.3. Crossover and Mutation Operations
Crossover and mutation operations in genetic algorithms simulate the process of genetic inheritance in nature, and crossover and mutation operations are the core operations that genetic algorithms can play. In this paper, the two genetic vectors of the parental path are \( \vec{a} \) and \( \vec{b} \), and the two progeny gene vectors generated by the cross operation are \( \vec{c} \) and \( \vec{d} \). The relationship between them is as follows:

\[
\begin{align*}
\vec{a} &= (a_1, a_2, a_3, a_4, \ldots, a_n) \\
\vec{b} &= (b_1, b_2, b_3, b_4, \ldots, b_n) \\
\vec{c} &= \vec{a} + \alpha \times (\vec{b} - \vec{a}) \\
\vec{d} &= \vec{b} + \alpha \times (\vec{a} - \vec{b})
\end{align*}
\]

\( \alpha \) is a randomly generated crossover coefficient, the range is a real number between 0 and 1. After the cross operation, it is necessary to perform the mutation operation on the progeny vector by a certain probability. As shown in Figure.5, the mutation operation randomly selects two different dimensions of the gene vector, and then exchanges the values of the two dimensions. Note that the progeny vector generated after cross-operation and mutation operations may not be reasonable because the solution to the problem corresponding to these progeny vector may not be able to satisfy the order constraint between the unit segments, so the above- The relaxation of the gene vector is modified to rationalize the progeny vector.

\[ \begin{array}{cccc}
0.1 & 0.2 & 0.2 & 0.3 \\
0.1 & 0.25 & 0.15 & 0.35
\end{array} \]

**Figure.5** Mutation Operation

5.4. Selection Process
The selection process in the genetic algorithm simulates the process of selecting the survival of the fittest. Selection of operations can transfer the parent's excellent genes directly to the next generation, or generate new individuals through the cross mutation operation and then to the next generation. In this
paper, the selection process is used to select the \( K \) gene vectors randomly from the population, and then find the maximum adaptability from the \( K \) gene vectors. The process is shown in Figure.6. The reason for this choice is because the method is simple and computationally efficient.

![Figure.6 Selection Process](image)

5.5. Genetic Algorithm to Solve the Process
After completing the definition of the coding and decoding of the gene, the fitness function, the cross mutation operation and the selection operation, the whole algorithm is solved as follows:

1. Using the random function, the initial parent population of scale \( N \) is randomly generated and the gene vector in the population is rationalized.
2. Calculate the fitness values of all the genetic vectors in the parent population.
3. Using selection, crossover and mutation operations, generates a new generation of population.
4. If the number of the progeny population reaches the set upper limit, the flow is stopped and the corresponding vector of the current optimal fitness value is output. Otherwise the new progeny population as a parent population, repeat the process (2) to (4).

In this case, the genetic algorithm is used to solve the TSP problem with sequential constraints. Under normal circumstances, the solution has been basically able to meet the needs of engineering practice after iteration.

6. Example verification
Take the parts shown in Figure.6 as an example. The trajectory of the empty stroke before optimization is shown in Figure.7. The optimized tool path is shown in Figure.8.

![Figure.7 the sample part](image)
By comparing the tool trajectories before and after the optimization, it can be seen intuitively that the order of the processing trajectory becomes more reasonable and the distance of the empty stroke is greatly shortened.

7. Conclusion
In this paper, we analyze the essence of the optimization of the idle stroke in the milling process. The Z coordinate value of the safety plane is automatically obtained by using the sampling point information of the cutting trajectory, and then the cutting track segment is divided into a section of the unit processing section. By using Z-Map simulation technique, the order constraints between the processing segments of each unit are obtained, and then the problem is transformed into TSP with sequential constraints, and then the genetic algorithm is used to optimize the solution.

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

References
[1] Yongzhi Pan, Xing Ai, et al. Tool geometry parameters and cutting parameter optimization based on the cutting force prediction model [J]. Chinese Journal of Mechanical Engineering, 2008, 19(4):428-431
[2] Haijiang Liu, Wei Huang. Optimization of nc machining parameters based on particle swarm optimization [J]. Journal of tongji university natural science edition, 2008, 36(6):803-806
[3] Zhengtao Huang, Jie Yang, et al. Modeling and parameter optimization of NC milling process for energy consumption [J]. Chinese Journal of Mechanical Engineering, 2016, 27(18):2524-2532
[4] Hsu P L, Yang W T. Realtime 3D simulation of 3-axis milling using isometric projection[J]. Computer-Aided Design, 1993, 25(4):215-224.
[5] Maeng S R, Choi B K, Baek N, et al. A Fast NC Simulation Method for Circularly Moving Tools in the Z-map Environment[C]// Geometric Modeling and Processing. 2004:319-328.
[6] Tao Z. TSP Problem Solution Based on Improved Genetic Algorithm[C], International Conference on Natural Computation. IEEE Xplore, 2008:686-690.
[7] Xiaoming Dai, Ruimin Zhou, et al. Hybrid parallel genetic algorithm is used to solve the TSP problem [J]. Journal of Electronics Information Technology, 2002, 24(10):1424-1427.
[8] Youyun Ao, Hongxin Chi. An algorithm for solving TSP problem based on genetic algorithm [J]. Computer & Digital Engineering, 2006, 34(4):52-55.