Society of Production Engineering

SPMS 2023
39. Savetovanje proizvodnog mašinstva Srbije
ICPES 2023
39th International Conference on Production Engineering of Serbia

Faculty of Technical Sciences
University of Novi Sad

Novi Sad, Serbia, 26. – 27. October 2023

INTEGRATED OPTIMIZATION OF TOOL PATH AND CUTTING PARAMETERS IN CONTOUR MILLING USING GENETIC ALGORITHM

Predrag MITIĆ[1]* , Aleksandar ĐORĐEVIĆ[2], Suzana PETROVIĆ SAVIĆ[2], Dragan DŽUNIĆ[2], Vladimir KOČOVIĆ[2]
[1]Metalika SMTR, Sopot, Serbia
[2]Faculty of Engineering, University of Kragujevac, Serbia
*predrag2904@gmail.com

Abstract: *The optimization of the cutting parameters and the tool path has attracted the attention of a large number of researchers in the past two decades. Cutting parameters and the tool path have a great influence on the production time and the quality of the machined surface, and therefore on the production costs. Optimization of cutting parameters and tool path are still the main directions of research in the field of machining process optimization. However, in the majority of research, the optimization of cutting parameters and tool path is done independently and their mutual influence is ignored. This paper discusses the possibility of integrated optimization of tool path and cutting parameters on the example of rough contour milling using genetic algorithm.*

Keywords: *Integrated optimization, tool path, cutting parameters, genetic algorithm*

1. INTRODUCTION

The problem of tool path generation and optimization in chip removal machining processes has attracted the attention of a large number of researchers for several past decades. In addition to the optimal tool path it is necessary to determine the optimal cutting parameters while respecting the multiple technological constraints imposed by the machine, the tool and the part geometry in order to achieve optimal part quality in minimal machining time. Machining in general and milling in particular is one of the main production processes used to manufacture durable goods. The cost optimization of production processes remains one of the major focus points of machine builders world-wide [1].

The conventional ways of selecting the tool path and the cutting parameters or programming the NC code used data from machining handbooks and the knowledge of programmer for optimal processing. However, the conventional NC programming has many disadvantages for instance increasing time and cost production and, decreasing accuracy and quality of the work piece. Modern CNC machine programming relies on commercial CAM software that speed up and facilitate CNC programming but the price of these software is high and to work with them, a highly skilled workforce is needed. These software's, however, do not offer optimization of the tool path and cutting parameters, but rely on empirical data and built-in, ready-made algorithms. In the literature there is a growing trend of development and application od models for automatic generation and tool path optimization and also for cutting parameters optimization. The developments of such models will not only shorten the reaction time of manufacturing system, but also improve the machine productivity through optimal selection of cutting parameters [3]. The most common approach is the application of artificial intelligence and metaheuristic algorithms, but there are very few works that simultaneously optimize tool path and cutting parameters. Although the tool path generation and optimization and the cutting parameters selection and optimization can be performed independently, the mutual influence of the tool path and cutting parameters on the production costs should not be ignored. There are three major issues in the process of integrated optimization of tool path and cutting parameters:

- recognition of geometric information of the work piece
- selection and optimization of cutting parameters
- tool path generation and optimization

Assuming that the geometry of the work piece is already recognized in a suitable manner, it is necessary to integrate activities that in modern CAM systems are implemented through interaction with the knowledge of programmers and as mentioned above, built-in, ready-made algorithms. Also, the programmer or operator on the machine enters the cutting depth, the number of revolutions of the main spindle, the speed of the auxiliary movement, the place where the tool starts moving as well as the place where the tool exits the machining zone, the clamping method, the position of clamping accessories and similar.

The aim of this paper is to present the theoretical framework of the integrated optimization of the tool path and cutting parameters mode as a basis for the automatic generation of NC code, that is, technological procedures in the narrower sense.

2. DEVELOPMENT OF MODULE FOR THE INTEGRATED OPTIMIZATION OF CUTTING PARAMETERS AND TOOL PATH

The basis for making parts by machine processing is the technical documentation (technical drawing, 3D model of the part being processed, etc.) and also the initial shape of the raw material from which the part is made. Figure 1 show a typical industrially inspired part in 3D model representation.

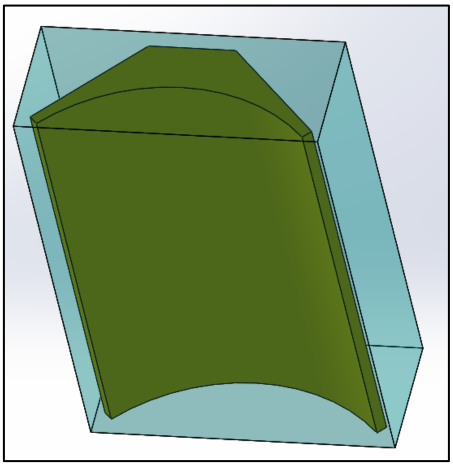

Figure 1. 3D model of work piece

Figure 2 shows a plane view of the same part, where the hatched surface represents the material that must be removed in order to obtain the shape marked in red, i.e. the final contour of the part. The shape and dimensions of the initial work piece (marked blue in Figure 1) is predetermined and as such, represents one of the input data in the model.

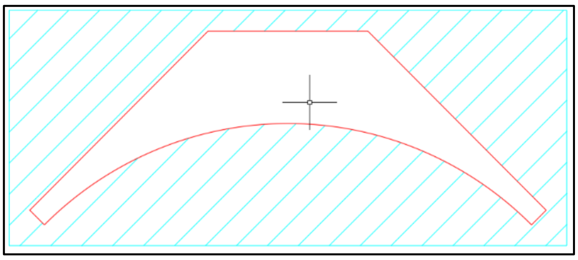

Figure 2. 3D model of work piece

In order for the part to be machined it is necessary to determine the tool, the cutting parameters (speed, feeds, radial and axial cutting depth) and the tool path. The cutting parameters and tool path must be determined in such a way that there is an economic profitability of the machining process, that is, their optimization must be performed in accordance with the optimization objectives and constraints imposed by various limitations of the tool, the CNC machine, the clamping tools, work piece material, the required accuracy and the required quality of the processed surface. Formally, the problem can be described as stated in [4]: For a work piece with given machine, cutting tool and clamp, there are a large number of feasible combinations $P=\{p_1, \ldots, p_i, \ldots, p_m\}$ of cutting parameters including spindle speed n, feed per tooth fz, cutting depth a_p and cutting width a_e. Each combination p_i corresponds to numbers of feasible tool paths $L_i=\{L_{i1}, \ldots, L_{ij}, \ldots, L_{in}\}$, where each tool path L_{ij} consists of void tool path L_a and cutting path L_c. In addition, cutting parameters and tool path affect each other and constitute processing space F. Then, the problem is defined as: for a given processing space F of a work piece, this paper aims to optimize the combination of P_i and L_{ij} to achieve coordinative optimization of objectives set O which can be expressed as:

$$\mathrm{F}_{\mathrm{optimal}} = \operatorname{argmin}\{O(P_i, L_{ij})\} \qquad (1)$$

Optimization objectives will be discussed later in the paper. The initial assumptions of the model are:

- the integrated optimization is done after the machine, cutting tool and clamping fixtures are chosen
- there are no tool change in the machining process of contour milling

It is clear that the process of integrated optimization of cutting parameters and tool path must take place in three stages. The first stage should be optimization of cutting parameters, the second stage is generation and optimization of the tool path and the final stage is interaction between optimized cutting parameters and the tool path. Figure 3 shows block diagram od proposed concept.

In [2] the authors stated that determining the optimal machine tool path has been proven to lead to high productivity and minimal production costs. Furthermore, they provide statistical data related to the methods used by various authors in researches, with the conclusion that Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) method are largely used to optimize machining efficiency and when compared with other methods, the GA has been successfully applied for many optimization problems with various parameters related to tool path and effective in improving the robustness of feature selection over a range of problems. A similar analysis of methods used to optimize cutting mode parameters is given in [4], so GA was chosen as the most suitable method for the research presented in this paper as well.

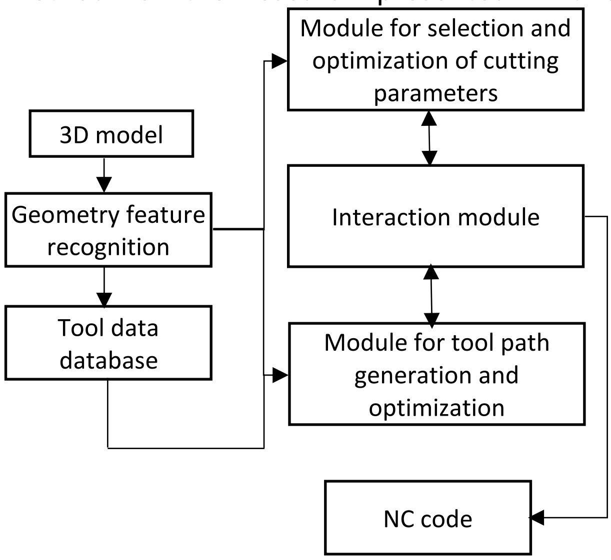

Figure 3. Block diagram of integrated optimization

2.1 Module for selection and optimization of cutting parameters

The task of optimizing the contour milling machining is defined at the following way: select the milling parameters speed and feed that meet the quality limits surfaces and optimization criteria [5]. The traditional methods for milling parameters calculation are widely known and are well described in the literature. Although the mentioned methods are applicable in everyday practice, nowadays, especially in the conditions of individual and small-batch production, data from the tool manufacturer's catalogue are most often used to determine the tool and milling parameters for the selected cutter. These catalogues are available in electronic form, so determining the tool and parameters using this data is very easy and fast. Based on the geometry of the surface and the required quality of processing, a tool of the appropriate diameter is chosen in such a way that all segments of the contour can be processed with that tool. The manufacturer of the tool according to the type of engagement (Figure 4) and according to the recommended values of the axial (a_p) and radial (a_e) depth of cut gives the recommended values of the milling parameters.

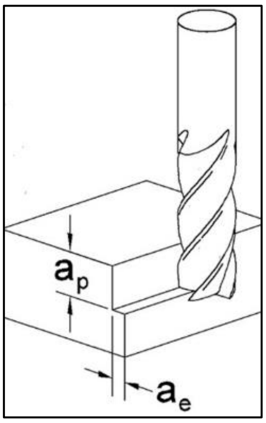

Figure 4. Tool engagement

The recommended values of cutting parameters are always given in the interval from the minimum to the maximum as shown in Table 1 as an example. In addition to data shown in Table 1 there are also recommended values for the axial (a_p) and radial (a_e) depth of cut in function od tool diameter D. Radial depth of cut a_e or *stepover* can be also defined as the percentage of the tool diameter engaged in material (Figure 5).

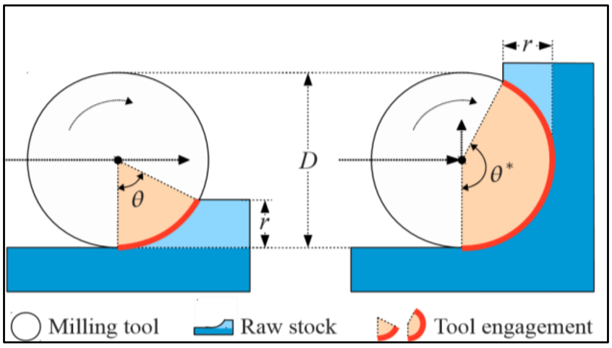

Figure 5. Radial depth of cut and TEA

The stepover determines the material removal rate (MRR) and reflects the cutting forces, but only for straight line motions. A parameter which better reflects the cutting force, regardless of the toolpath shape, is the *tool engagement angle-TEA* [6] as the amount of sweep subtended by each cutting edge as it engages and leaves the stock. TEA plays an important role in the formation of the tool path and is dependent on the axial depth of cut [5]. The engagement angle reaches its maximum (360°) when plunging the tool vertically into the material. The next maximum value, 180°, is encountered during a slotting operation; this condition may lead to high thermal stress on the tool, since the chips cannot be evacuated properly. Tool engagement is also known to increase at internal corners in toolpath. The engagement angle also has direct influence on the chip shape, therefore keeping TEA constant ensures consistent chip size and shape throughout the milling process [6]. A large value of the TEA would result in a large amount of material removed, which certainly increases productivity, but then the cutting resistance is high and the wear of the tool is intense. In the model presented in this paper TEA will be considered in the process of forming feasible tool paths as one of the constraints. The maximum value of TEA marked as Θ_{max} can be easily calculated from the data shown in Table 1 and Table 2 and is often found in tool manufacturers' catalogs as a value given in degrees.

Input values for GA that generates sets of cutting parameters, chosen from a tool manufacturers' catalog are:

- tool diameter D (mm)
- number of teeth *z*
- cutting speed V_c (m/min) within the limits $V_{cmin}<V_c<V_{cmax}$
- feed per tooth f_z(mm/tooth) within the limits $f_{zmin}<f_z<f_{zmax}$
- radial depth of cut a_e (mm) within the limits $a_{emin}<a_e<a_{emax}$
- axial depth of cut a_p (mm) within the limits $a_{pmin}<a_p<a_{pmax}$

Also, maximum TEA defined as Θ_{max} is considered as maximum possible value for above cutting parameters values within the given limits. The above given limits of cutting parameters are constraints of the presented model and must be included in GA through chromosome feasibility check. In addition to the above mentioned constraints, it is necessary to take into account the constraints related to tool shaft breakage (cutter bending strength), tool deflection and the speed of main spindle.

Table 1. The recommended value of cutting parameters

				fz [mm/tooth] at Diameter				
	Strength [N/mm^2]	**Work piece material DIN**	**Vc [m/min]**	**2-4**	**4-8**	**8-12**	**12-16**	**16-20**
1. Work piece material								
1.1 Free cutting steel	< 900	9 S 20	220-230	0.03-0.04	0.04-0.09	0.09-0.13	0.13-0.18	0.18-0.22
1.2 Structural steel	<500	ST 37-2	220-230	0.03-0.04	0.04-0.09	0.09-0.13	0.13-0.18	0.18-0.22
1.3 Structural steel	> 500	ST 60-2	190-200	0.025-0.035	0.035-0.08	0.08-0.12	0.12-0.16	0.16-0.02
1.4 Tempered steel	<1000	42 CrMo 4	150-160	0.025-0.035	0.035-0.08	0.08-0.12	0.12-0.16	0.16-0.02
1.5 Cast steel	<1000	GS-45	120-130	0.02-0.03	0.03-0.07	0.07-0.1	0.1-0.14	0.14-0.17
1.6 Case-hardened steel	<1200	16 MnCr 5	190-220	0.02-0.03	0.03-0.07	0.07-0.1	0.1-0.14	0.14-0.17
1.7 Stainless steel ferritic/martensitic	<1100	X 10 Cr 13	110-120	0.01-0.02	0.02-0.04	0.04-0.06	0.06-0.08	0.08-0.1

The calculation of maximum cutter bending stress and tool deflection, in detail, can be found in [3] and must be incorporated in GA mechanism as constraints during feasibility check. Using wide known formula for spindle speed

$$\mathrm{n}\ (o/min) = \frac{V_c \cdot 1000}{\pi \cdot D} \tag{2}$$

it is easy to determine n_{max} and n_{min} using V_{cmin} and V_{cmax} obtained from Table 1. Now, all constraints of the presented model are defined.

In GA acceptable representation of chromosome is the most critical factor influencing all other phases of the GA [7]. In the present model a single chromosome binary bit string is chosen to represent four cutting parameters as shown in Table 2. The number of bits in a chromosome which represents coded chromosome size is set to 24. Cutting parameters are mapped from the 24-bit chromosome by segmenting it into four equal parts.

Table 2. Chromosome representation

1	0	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1
ap (mm)						*ae (mm)*						*n(o/min)*						*fz* (mm/tooth)					

Each part of the chromosome string represents a percentage value of the predefined range of the cutting parameter and is obtained using equation (3) [3]. The maximum value for each part is 63, which is value obtained from binary to decimal conversion of maximum value, and the value of zero correspond to the upper and lower bounds of the cutting parameter, as shown in Table 1.

$$X = \frac{X_{max} - X_{min}}{63} \cdot Y + X_{min} \tag{3}$$

where X is mapped value of the cutting parameter, X_{max} and X_{min} correspond to the cutting parameter's upper and lower bound, while Y is the decoded value of respective chromosome part.

Furthermore, an algorithm for forming a set of parameters, based on a GA, will be given. For every chromosome, Material Removal Rate (MRR) is calculate given by an equation (4)

$$\max(MRR) = \max\left(f_z \cdot z \cdot n \cdot a_p \cdot a_e\right) \tag{4}$$

A set of cutting mode parameters is formed, which is passed to the interaction module, which will be discussed later. The algorithm is repeated *m* times, and *m* is the value that represents the input data to the algorithm for integrated optimization. As the mechanism of action of GA is widely known [7], in this part only specific details related to the algorithm for forming a set of parameters will be given.

<u>Step 1:</u> Entry of input parameters-population size, no. of generations, no. of parents, mutation probability, crossover probability

<u>Step 2:</u> Randomly generation of a bit binary string-chromosome

<u>Step 3:</u> Checking feasibility of chromosome. If chromosome is feasible then go to step 4. If not go to step 2.

Step 4: Calculate MRR of chromosome

<u>Step 5.</u> Store chromosome p_i in the set P

<u>Step 6:</u> Checking if population size is equal to the cardinality of set P. If it equal go to step 7. If not go to step 2

<u>Step 7:</u> Transmit the set P to the interaction module via data in data interface

Now, the set of feasible chromosome is formed which representation the set of cutting parameters with MRR calculated for each of them.

2.2 Module for tool path generation and optimization

In contour milling, the tool path represents an ordered set of points where the tool is positioned, moving in the direction of the auxiliary motion velocity vector. If *n* is the number of points in that set, then theoretically there are *n!* potential toolpaths. If restrictions are taken into account, then that number decreases but is still very large. Discretization can be modelled as pixelization-mapping of the surface being processed with a large number of squares [8], basically cover the surface being processed very well, but then their number, that is, the resolution of the grid of squares goes up to 0.01 mm is the usual resolution of today's CNC machines. Such a data set is very large for processing by metaheuristic methods, especially GA, which can lead to the fact that it is not possible to find a solution that converges to the optimal one in an acceptable time [9]. Other approach is surface discretization by placing up an equidistant grid of points on the surface to be processed, but this method gives good results only in cases of rough contour milling [10]. For the purpose of integrated optimization od tool path and cutting parameters discretization of the surface is done by placing points at a distance equal to the radial cutting depth a_e which is part of the set of parameters of the cutting mode. Example of discretization is shown in Figure 6.

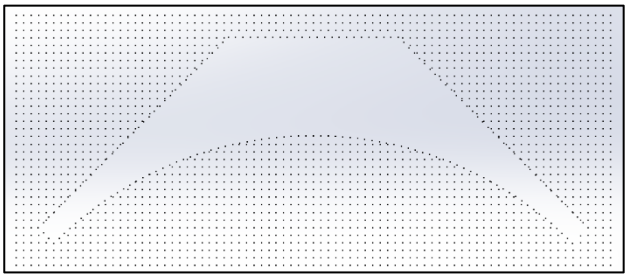

Figure 6. Surface discretization

Distance between points in point grid is equal to a_e with the exception of the points located on the offset contour of the work piece. It is clear that contour of work piece is offset for half of tool diameter D. The algorithm for tool path generation and optimization can be found in [11] and here a brief description of the most important features will be given. It is based on, in the literature, widely known problem of the Traveling Salesman Problem (TSP) [7].

A 3 axes CNC machine moves in both *x* and *y* directions simultaneously and Euclidean distance function is to be used to calculate the distance between points. In this way a distance matrix D=[d_{ij}] between points (Figure 6) is created. Let M is a set of points obtained by surface discretization

Let *i* and *j* be two arbitrary points from set M.

<u>Input variables of the proposed model</u>

- the set of points described by the point map M= [m_{ij}],
- the distance matrix D=[d_{ij}] between the points of the set M
- maximum TEA θ_{max} in degrees

- tool diameter d in mm
- speed of main spindle n(rev/min)
- feed rate $F = f_z \cdot z \cdot n$ (mm/min)
- rapid feed rate F_{bh} (mm/min)

Control variables of the proposed model

$$X_{ij} = \begin{cases} 1 \; if point\; i\; is\; immediately\;\; by\; j \\ 0 \qquad\qquad otherwise \end{cases} \quad (5)$$

$$Y_{ijk} = \begin{cases} 1 \; if\; tool\; travels\; from\; i\; to\; k\; in\; same\; dir. \\ 0 \qquad\qquad otherwise \end{cases} \quad (6)$$

$$Z_{ij} = \begin{cases} 1 \; if\; j\; is\; machined \\ 0 \quad otherwise \end{cases} \quad (7)$$

Θ_{ij} *TEA when moving from i to j* (8)

Constraints of the proposed model

1) Constraint that ensure that every point from set M is visited by the tool at least once

$$\forall j \in \{0 \ldots N\} \sum^{N} X_{ij} > 0 \quad (9)$$

2) Constraint that ensure that tool leaves each point after visiting for another

$$\forall j \in \{0 \ldots n\} \sum_{i=0}^{n} X_{ij} \sum_{k=0}^{n} X_{jk} \quad (10)$$

3) Constraint that ensure that every point is followed by a different point and path moves on

$$\forall j \in \{0 \ldots n\} \, X_{ii} = 0 \quad (11)$$

4) From each subset of points the tool must be positioned at least once in another point that is not part of the of that subset of points

$$\forall Q \subset \{0 \ldots n\} \colon Q = \emptyset$$

$$\sum_{\{i,j \in N\} \colon i \in Q, j \in N \setminus Q} X_{ij} + \sum_{\{i,j \in N\} \colon i \in Q, j \in N \setminus Q} X_{ji} \geq 2 \quad (12)$$

5) Constraint to ensure that the tool engagement angle Θ_{ij} must be smaller or equal than the maximum allowed engagement angle Θ_{max}

$$\forall j i \{0 \ldots n\} \sum_{i=0}^{n} X_{ij} \cdot \theta_{ij} \;\leq \theta_{max} \quad (13)$$

6) Constraint that ensure that during the chip removal process, the movement of the tool is allowed only towards neighbor points.

$$\forall j \in \{0 \ldots n\} \sum_{i=0}^{n} Z_{ij} \cdot X_{ij} = 0 \quad (14)$$

The objective function

The objective function of the proposed model represents the criteria of optimization. In the observed model the goal is to achieve high productivity, thus minimization of processing time, with minimization of the jerk effect with as uniform tool angle of engagement as possible and also

Minimization of the processing time, can be written as:

$$F_{c1} = min \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{d_{ij}}{F} + \frac{r_{ij}}{F_{bh}} \right) \cdot X_{ij} \quad (15)$$

where d_{ij} are the distance between the points of the path in which the tool moves in a feed rate and r_{ij} the distance between the points of the path in which the tool moves in a rapid rate.

The minimization of the number of changes in the direction of movement of tool, taking into account the control variable defined by expression (6) can be written as:

$$F_{c2} = min \sum_{i=1}^{n-2} \sum_{j=2}^{n-1} \sum_{k=3}^{n} \frac{1}{Y_{ijk}} \quad (16)$$

The third goal of optimization is the smallest possible deviation of the TEA from the target value and can be defined as:

$$F_{c3} = min \sum_{i=1}^{n} \sum_{j=1}^{n} \left| \theta_{ij} - \theta_c \right| \cdot X_{ij} \quad (17)$$

that is, as a minimization of the deviation of the TEA value at each point of the path in relation to the target value.

The problem of milling path optimization is clearly multi objective optimization problem. For the purpose of this research the method of weight coefficients will be applied. Determining the weight coefficients can be a problem [7] because the vector of weight coefficients controls the optimal solution. Mathematically, the optimal solution obtained with equal weighting coefficients should lead to the smallest conflict between the optimization goals, but in practice it is often not a satisfactory solution, so when determining the weighting coefficients, it is always necessary to have information in the order of priority of the goals. For the purposes of this paper, the greatest weight will be given to achieving maximum productivity so objective function is defined by following expression

$$F_c = 0{,}5 \cdot F_{c1} + 0{,}25 \cdot F_{c2} + 0{,}25 \cdot F_{c3} \quad (18)$$

Chromosome representation

Now it is necessary to define an appropriate genetic representation or an appropriate coding method. In the model being observed so that the chromosome coding solution with a vector of real components will be given.

As mentioned earlier in this paper the geometry of working piece is already recognized. The coordinates of every point is known so the distance matrix D=[d_{ij}] can be easily determined. In the observed model, the tool can pass through any point of the map of points several times, with the fact that it only passes once to remove chips, and it can pass through the same point again only for the purpose of approaching the cutting zone. Thus, the gene in the chromosome is marked with a natural number and represents the ordinal number of the point of the point map through which the tool passes. The tool can move from one point to another either at feed rate or rapid rate depending on whether the chip is removing or tool positioning is performed. Therefore, it is necessary to know the speed with which the tool moves from the point *i* and arrives at the point *j*. The next segment that describes the trajectory of the cutter is the engagement angle θ_{ij} which can be $\theta_{ij} = 0$ if the tool is moving at rapid rate or $\theta_{ij} > 0$ if the tool is moving at feed rate. In addition to knowing the x and y coordinates of each path point, it is also necessary to know the z coordinate, i.e. the plane in which the center of the tool moves and the condition of the point i.e. if is previously machined o no. An example of fully decoded chromosome is given in Table 3.

Initial population

Using the distance matrix D=[d_{ij}], it is necessary to form another set of points which is very important for generating the initial population. That set K is the set of the closest points of each point of the point map, which contains all the k closest points to the point where the center of the tool is located. In the observed model, there can be up to 24 closest points to each point of the point map, and the x and y coordinates of the members of that set tell to which point the tool can move from any current point of the path. By forming the set K, a part of unacceptable or illegal paths is eliminated, i.e. the constraint defined by expression (13) is implemented. When all points of set K is visited by the tool then next random point of tool path is chosen. At the initial moment. only the points located in the first and last row and the first and last column of the map of points have the condition *machined=true* and that point are peripheral points status of peripheral points, all other points have the status *machined=false*, i.e. the status of unprocessed points. It is clear that the first point of any milling path must be one of the points with status *machined=true*. Furthermore, the algorithm for forming the initial population is given below:

Step 1: Randomly select the initial point from the set of points M with machined=true

Step 2: For a randomly selected point from step 1, load the previously determined set of nearest points K to the randomly selected point.

Step 3: Randomly choose one point from the set K in the label j that has the status of an unprocessed point i.e. *machined=false*

Step 4: Calculate the milling angle θ_{ij} when moving from i to j.

Step 5: If $\theta_{ij} \leq \theta_{max}$, point j becomes a tool path point. If $\theta_{ij} \geq \theta_{max}$max then go to step 10.

Step 6: Assign status *machined=true* at point j.

Step 7: Calculate the vector product of vectors and $\vec{i}x\vec{j}$ if the vector product is different from zero, increase the control variable t=t+1, if it is equal to zero, the control variable t keeps its previous value.

Step 8: Calculate $\left|\theta_{ij}-\theta_c\right|$. Control variable $d=d+\left|\theta_{ij}-\theta_c\right|$.

Step 9: Check the statuses of the neighboring points of point j. If the statuses are machined=true, then point j becomes a peripheral point, i.e. the status of point j machined=true

Step 10: Repeat steps 3-9 for each point from the set K.

Step 11: Repeat steps 1-10 for each point in the set M until all points from the set M have status machined=true

Step 12: Repeat steps 1-11 for each individual from the initial population

Table 1. Decoded chromosome

	1	3	8	5	4	2	10	7	9	8	3	5	6
F (mm/min)	F_{bh}	F	F	F	F	F	F	F	F	F_{bh}	F_{bh}	F	F
θ_{ij} (°)	0	40	42	25	30	35	39	41	42	0	0	0	30
x (mm)	20	22	24	26	28	30	32	32	34	24	22	26	26
y (mm)	6	6	6	6	6	6	6	6	6	6	6	6	10
z (mm)	50	0	0	0	0	0	0	0	0	50	50	0	0
Machined	true	false	false	false	false	false	false	false	false	true	true	false	false

By forming the initial population whose size is the input parameter of the genetic algorithm, the initial set of possible tool paths is formed. By choosing the parents, that is, two tool paths, and applying the genetic operators of crossover and mutation, and calculating fitness for each individual, that is, for each tool path, a path that converges to the optimal path is reached. Furthermore a pseudo code of proposed GA for tool path generation and optimization will be given. The detail description of fitness calculation, parents selection, crossover and mutation operator as mentioned can be found in [11].

Input parameters for GA are: Set of point M, TEA target value, diameter of the tool, the population size, the number of parents, mutation rate and the number of generations.

Start
Enter the input parameters
Create Points map, Distant matrix and Matrix of neighbors K of each point
Create Initial population
Calculate the fitness of individuals of the first generation
    Generation = 1
        Repeat
            Selection of parents
            Children = 0
            Repeat
                Pick of two parents for a crossover (Parent1, Parent2)
            Crossover OX (Parent1, Parent2)
            Mutation
            Children = Children + 1
            Until Children = population size-total number of parents
            Generation = Generation + 1
            Calculate the fitness of individuals for the current generation
Until generation = total number of generations
End.

In the next section algorithm for interaction between cutting parameter optimization and tool path generation and optimization will be shown.

2.3 Interaction module

In this chapter, a two-stage algorithm based on analysis presented in [4], will be given that combines a GA for cutting parameter optimization and a GA for toolpath generation and optimization.

Step 1: Setting the parameters of both algorithms which includes population size, maximum number of iterations *m*, crossover probability, mutation probability, number of parents and number of generations.

Step 2: Execution of cutting parameters optimization algorithm which generates set of cutting parameters.

Step 3: Transmission of a set of generated cutting parameters to tool path optimization module via data interface.

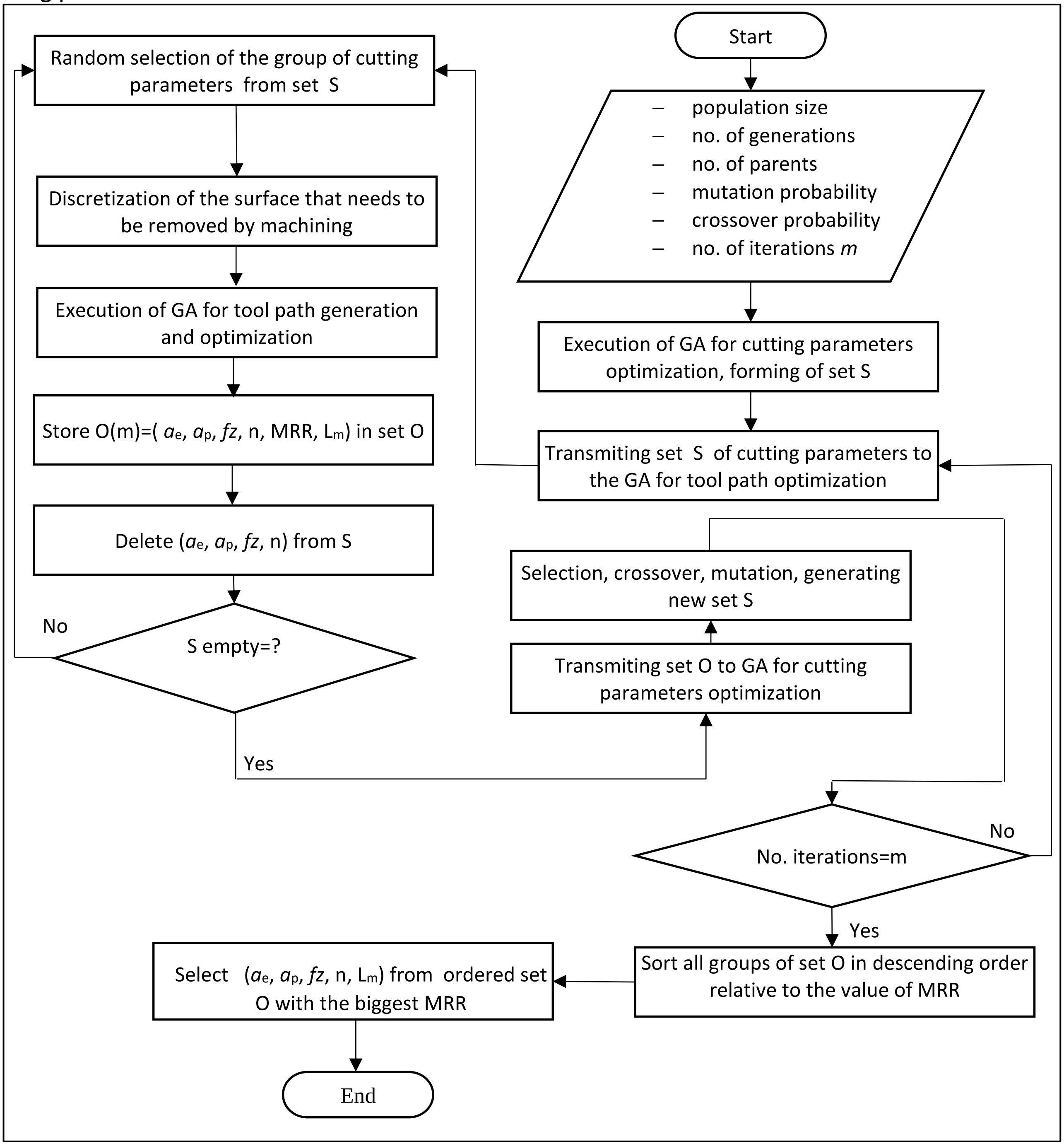

Figure 7. Flowchart of proposed algorithm for integrated optimization

Step 4: One group (a_e, a_p, fz, n) of cutting parameters is randomly selected from the set P.

Step 5: Discretization of the surface that needs to be removed by machining, by imposing a grid of points, based on the selected radial depth of cut a_e

Step 6: Execution of tool path generation and optimization algorithm, which give the optimum tool path for used set of cutting parameters.

Step 7: The group of cutting parameters with MRR and optimum tool path is stored in set O

Step 8: The extracted group of cutting parameters is deleted from set P

Step 9: Is set P empty=?

Step 10: If yes, optimum objective set O is transmitted to the cutting parameters module via data interface, if not go to step 4.

Step 11: In the cutting parameters module a series of operations is conducted, including selection, crossover and mutation to obtain new population of cutting parameters.

Step 12: Is reached maximum number of iterations for cutting parameters optimization algorithm=?

Step 13: If yes, then the process of generation the optimum objective set O (group of cutting parameters and the tool path) is finished, if not go to step 3.

Step 14: Sort all groups of set O in descending order relative to the value of MRR

Step 15: The first group of ordered set O is selected as the optimal set of cutting mode parameters with the corresponding tool path.

Step 16: Information about optimal set of cutting parameters and corresponding tool path is transmitted to the module for generating NC code via data interface

For better understanding of proposed algorithm for integrated optimization a flowchart is given in a Figure 7. The basic idea is that the set of groups of the cutting parameters is formed several times, that is, that more populations of individuals are formed, which would significantly increase the diversity of population. Each of those individuals is sent to the tool path generation and optimization module, which increases the diversity of the individuals represented by the tool paths, and the information about the best tool path for the corresponding group of cutting parameters is sent to the cutting parameter optimization module, where by applying elitism can be selected the best individuals or groups of cutting mode parameters in the next iteration. In this way, the diversity of individuals in the GA is preserved, but this prevents its premature convergence towards the optimal solution, and at the same time bad individuals are eliminated from the population.

3. CONCLUSION

The classic method of creating NC programs relies on CAM software and the experience of the programmer. Even then, the time of creating the program can be long and there is no guarantee that the machining time will be as short as possible which is necessary condition for minimizing the production costs. In the recent decades, there has been a growing trend in the literature in developing models for tool path generation and optimization as well as cutting parameters optimization in order to automate the process of NC code generation. These problems are being observed separately, regardless of the fact that the cutting parameters affect the generation of the tool path. This research deals with the possibility of integrated optimization of cutting parameters and tool path generation and optimization. For this purpose, the theoretical framework of integrated optimization as well as the algorithm that integrates the optimization of cutting parameters and tool path are given, as a prerequisite for the automatic generation of the NC code, that is, the technological procedure in the narrower sense.

REFERENCES

[1] D. Khatiwada, D. Nepali, N. Raj, A. Bhattarai: Tool path optimization for drilling holes using genetic algorithm, International Journal of Machine Tools and Maintenance Engineering, 1(1):36-42, 2020

[2] S. Bharath, K.R. Natraj: Application of Artificial Intelligence Methods of Tool Path Optimization in CNC Machines, International journal of engineering research and *technology,* Volume 06, Issue 14, 2018

[3] J. Rai. D. Brand, M. Slama. P. Xirouchakis: Optimal selection of cutting parameters in multi-tool milling operations using a genetic algorithm. International Journal of Production Research, Volume 49, No. 10, 3045-3068. 2011

[4] S. Jia, S. Wang, N. Zhang et al.: Multi-objective parameter optimization of CNC plane milling for sustainable manufacturing, Environmental Science and Pollution Research,

https://doi.org/10.1007/s11356-022-24908-3, 2022

[5] Lj. Tanović, Y. Petrakov: Teorija i simulacija procesa obrade, *Mašinski fakultet*, Beograd, 2007

[6] A. Dumitrache, T. Borangiu, A. Dogar: Automatic Generation of Milling Toolpaths with Tool Engagement Control for Complex Part Geometry, in: *IFAC Proceedings,* Volumes, 43, 252-257, 2010

[7] S. R. Singiresu: Engineering optimisation-theory and practice, Fourth edition*, John Wiley & Sons, Inc.*, Hoboken, New Jersey, 2009

[8] M. Kumar, P. Khatak: Implementation of Genetic Algorithm in Multiobjective Optimization of Milling Toolpath, International Journal of Emerging Technologies and Innovative Research*, ISSN:2349-5162,* Vol.5, Issue 5, page no.1147-1155

[9] J. Barclay, V. Dhokia, A. Nassehi: Generating milling tool paths for prismatic parts using genetic programming. *Procedia CIRP, 33*, 490-495. 2015

[10] W. Essink, A. Nassehi, S. T. Newman: Toolpath Generation for CNC Milled Parts Using Genetic Algorithms*,* Enabling Manufacturing Competitiveness and Economic Sustainability (pp.189-193)*,* doi:10.1007/978-3-319-02054-9_32, 2015

[11] P. Mitić, M. Zahar Đorđević, V. Petronijević, N. Abadić. A. Đorđević: Automatic tool path generation in contour milling using genetic algorithm, *14th International Quality Conference, Quality Festival 2023,* Kragujevac, 2023, 24-27 May, pp. 663-680, ISBN 978-86-6335-104-2