machines

MDPI

Article

MBD-Based Machining Feature Recognition and Process Route Optimization

Shuhui Ding [1,2], Zhongyuan Guo [1], Bin Wang [1], Haixia Wang [1,*] and Fai Ma [2]

[1] College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266590, China

[2] Department of Mechanical Engineering, University of California, Berkeley, CA 94709, USA

* Correspondence: wanghx@sdust.edu.cn

Abstract: Machining feature recognition is considered the key connecting technique to the integration of Computer-Aided Design (CAD) and Computer-Aided Process Planning (CAPP), and decision-making of the part processing scheme and the optimization of process route can effectively improve the processing efficiency and reduce the cost of product machining cost. At present, for the recognition of machining features in CAD models, there is a lack of a systematic method to consider process information (such as tolerance and roughness) and an effective process route optimization method to plan part processing procedures. Here we represent a novel model processing feature recognition method, and, on the basis of feature processing plan decision, realize the optimization of the process route. On the basis of a building model Attributed Adjacency Graph (AAG) based on model geometry, topology, and process information, we propose an AAG decomposition and reconstruction method based on Decomposed Base Surface (DBS) and Joint Base Surface (JBS) as well as the recognition of model machining features through Attributed Adjacency Matrix-based (AAM) feature matching. The feature machining scheme decision method based on fuzzy comprehensive evaluation is adopted, and the decision is realized by calculating the comprehensive evaluation index. Finally, the Machining Element Directed Graph (MEDGraph) is established based on the constraint relationship between Machining Elements (MEs). The improved topological sorting algorithm lists the topological sequences of all MEs. The evaluation function is constructed with the processing cost or efficiency as the optimization objective to obtain the optimal process route. Our research provides a new method for model machining feature recognition and process route optimization. Applications of the proposed approach are provided to validate the method by case study.

Keywords: machining feature recognition; process route optimization; machining elements directed graph (MEDGraph); improved topology sorting algorithm

check for updates

Citation: Ding, S.; Guo, Z.; Wang, B.; Wang, H.; Ma, F. MBD-Based Machining Feature Recognition and Process Route Optimization. *Machines* **2022**, *10*, 906. https://doi.org/10.3390/machines10100906

Academic Editor: Pingyu Jiang

Received: 16 September 2022
Accepted: 5 October 2022
Published: 8 October 2022

1. Introduction

Automatic Feature Recognition (AFR) refers to the extraction of feature information with specific engineering significance from part models. A CAD model of a part is composed of low-level graphic elements such as surface, line, and point, while CAPP information is generated based on high-level features with engineering semantics. Feature recognition is an intelligent interface between CAD and CAPP. AFR can identify the corresponding features according to specific process requirements, which is the basis of process planning and the most important step in transforming design information into manufacturing process information. With the help of AFR, the structure feature can be identified according to specific process information, which is considered the most crucial technique to turn design information into a manufacturing process [1].

In recent years, feature recognition has developed rapidly, but the traditional feature recognition methods based on model topology and surface geometry recognition do not take model process information into account. If the roughness and tolerance grade of the assembly surface of two models with identical topology and geometric dimensions are quite different, their processing technology may be completely different. Therefore, machining feature recognition based on machining information has become a problem that must be considered.

Model-Based Definition (MBD) is a model-based product data definition method that adds tolerance, annotation, and other process information on the basis of model topology and geometric data. Based on the geometric and process information in the MBD model, the processing feature recognition based on MBD lays a foundation for the subsequent feature processing scheme decision and process route optimization.

The paper plans to solve the problems of machining feature recognition, machining scheme decision, and process route optimization in the MBD model. On the basis of extracting the model geometry and process information, the model Attributed Adjacency Graph (AAG) is established and decomposed to obtain a feature subgraph, and then the feature subgraph is matched with predefined machining features to realize machining feature recognition. The decision-making method of the model feature processing scheme based on fuzzy comprehensive evaluation is adopted. By analyzing the influencing factors of the feature processing scheme, the subordinate relationship between each influencing factor and the processing scheme is established. A multi-factor comprehensive evaluation model is established according to the single-factor evaluation results, and the optimization of feature processing schemes is realized by calculating the comprehensive evaluation index vector. Finally, based on the constraint relationship between the processing elements, a Machining Elements Directional Graph (MEDGraph) is established. The improved topological sorting algorithm lists all the topological sequences of the processing elements, takes the processing cost or efficiency as the optimization objective, constructs an evaluation function, and obtains the optimal process route.

The remainder of the paper is organized as follows. Section 2 reviews the literature relating to feature recognition. In Section 3, the method of information extraction from MBD is introduced. Model AAG-based machining feature recognition method is constructed in Section 4. In Section 5, the feature processing scheme is decided. In Section 6, the model process route is optimized. The proposed methods are validated through a case study in Section 7, followed by a conclusion in Section 8.

2. Literature Review

In order to achieve feature recognition and process route optimization based on the CAD model, scholars carried out research in the fields of machining feature recognition and process route optimization.

2.1. Machining Feature Recognition

The purpose of machining feature recognition is to realize the conversion from design features to machining features. The concept of feature recognition was first proposed by Kyprianou [2], and its main research object is the geometric topology classification of models. Since then, various types of feature recognition methods have emerged, including graph-based methods, volume decomposition-based methods, neural network-based methods, trace-based methods, rule-based methods, and hybrid methods combining various methods [3].

The graph-based method is considered to be the most successful method. This method describes the feature recognition problem as a graph matching problem. The B-rep model is converted into a graph representation. Joshi and Chang first proposed the feature recognition method of AAG [4]. By acquiring the geometric topological relationship between surfaces and edges of the model, AAG is constructed, and then the machining features are identified by subgraph matching. However, in its research, only the concave–convex attributes of surfaces and edges are considered, and the model information is not fully expressed. Li [5] adds other attributes of vertices on the basis of AAG to form an Extended Attributed Adjacency Graph (EAAG). Marefat and Kashyap proposed the concept of virtual connection and tried to recover the boundary information lost due to the intersection of features [6]. This method realized the recognition of intersection features. However, the virtual connection construction is difficult, and the method is not universal.

Armstrong first proposed feature recognition based on unit decomposition [7]. Sakurai et al. conducted more in-depth research and made great progress [8–10]. This method can effectively identify intersection features and provide multiple interpretations [11]. However, this method is low efficient, so it needs to be further filtered. Campana and Mele proposed a rule-based feature recognition method [12], which needs to formulate predefined rules for each type of feature to be recognized. However, the definition of feature rules is not unique, and the calculation of feature recognition methods is large. Requicha et al. proposed a feature recognition method based on traces [13], which mainly studies the traces generated when different features intersect inside the part. Sunil and Pande proposed a feature recognition method based on neural networks [14], which can effectively recognize intersection features. However, due to the problem of data source, the accuracy of its recognition results cannot be guaranteed.

Hybrid feature recognition methods combine the advantages of two or more traditional feature recognition methods. Li et al. used the hybrid feature recognition method to improve the recognition reliability of rib features [15]. Gao and Shah proposed an intersection feature recognition method based on the mixture of graph and rule [16]. Fougeres and Ostrosi adopted a hybrid representation combining framework, syntax, and diagrams for the representation of features [17].

Liu et al. proposed a hybrid feature recognition method based on graph decomposition and feature factor clustering [18]. Zhang et al. input the geometric topology information of the model into the improved BP neural network for feature recognition by reading the step neutral file [19]. Duan et al. proposed a feature recognition method based on feature layering for cabin parts [20]. Luo et al. proposed a hybrid feature recognition method based on the combination of graph and rule-based reasoning [21].

The above feature recognition technologies have the defect of insufficient correlation between the identified features and model process information, which is not conducive to the processing of later process information.

2.2. Feature Processing Scheme Decision

Feature processing scheme decision-making is an important step in process design, which requires comprehensive consideration of many factors, such as material, type of processing features, and processing accuracy. The application of novel decision-making methods, such as the production rule decision-making method, method based on decision table or decision tree, fuzzy reasoning decision-making method, and neural network decision-making method, greatly improves the decision-making efficiency of the processing scheme compared with the traditional method based on process manual.

Berenji et al. proposed a decision-making method based on production rules to determine the processing scheme through the selection structure of judgment [22]. Huang established a mathematical model of fuzzy comprehensive evaluation for the decision-making of feature processing schemes and realized the decision-making process by synthesizing the fuzzy relations [23]. Ma et al. analyzed the fuzzy mapping relationship between feature attributes and the processing scheme and established a processing scheme decision model based on fuzzy reasoning [24]. Hu et al. established a feature processing scheme decision-making model based on Tsukamoto fuzzy reasoning and realized the optimization of feature processing schemes according to the membership degree of each processing scheme [25]. Zhong [26] and Zhao [27] used the neural network method to make feature processing scheme decisions. Guan et al. combined a genetic algorithm and Analytic Hierarchy Process (AHP) to realize the decision of the processing scheme [28]. Hu et al. used particle swarm optimization to optimize processing schemes with the goal of product benefits and product quality [29].

2.3. Process Route Optimization

The formulation of process route is the key to product production, which determines the machining process and cost of parts [30]. The traditional process route formulation method is inefficient and difficult to achieve optimization. With the development of computer technology such as artificial intelligence and data mining, the application of various intelligent algorithms makes it possible to formulate product process plans with high efficiency, low energy consumption, and low cost [31].

Chen et al. studied the CAPP system based on the feature parameter model and realized the process route decision by extracting the geometric feature parameters on the model and matching them with the template base [32]. Peng et al. extracted the feature factors of gear blanks using the fault tree analysis method and obtained the manufacturing process route with the best ecological efficiency [33]. Krishna et al. proposed a novel heuristic ant colony algorithm to search for the global optimal process route, constructed the optimal objective function model, and verified its progressiveness [34]. Salehi et al. proposed a two-level planning process route optimization method [35]. Zhou et al. built a multi-objective model to optimize the low-carbon-based process route optimization method, which determines the minimum carbon emissions and processing time as the objective of optimizing the process route [36].

Cheng et al. integrated the ant colony algorithm and bacterial foraging algorithm and designed two search phases of manufacturing resources and processing element sequence [37]. An et al. proposed an optimization algorithm combining intuitionistic fuzzy sets and the second generation of intensity Pareto to construct a multi-objective function with the lowest replacement cost of processing resources to realize the optimization of the process route [38]. Based on the genetic algorithm, Guo et al. introduced feasible process sequences to improve the genetic algorithm, and the convergence speed and global optimization ability of the genetic algorithm were improved [39–41]. Liu et al. combined polychromatic set theory and genetic algorithm and took the shortest auxiliary time as the optimization goal, realizing the work step sequencing in the machining center [42].

3. MBD-Based Model Information Extraction

MBD model data express product design and manufacturing information, including 3D model data, datum, coordinate system, annotation information, and other data, which can provide information support for subsequent process planning of products. Currently, MBD has become a general model definition method. Every mainstream 3D product design software has established a data conversion interface to STEP files and also developed corresponding customized development functions, such as Creo Toolkit of Creo, UG Open provided by UG, and CATIA CAA of CATIA.

The identification of model machining features mainly involves solid model information and annotation information in the MBD model [43]. The solid model reflects the product geometry and is composed of the basic geometric elements through topological connection, including model geometry information and auxiliary geometry information. Auxiliary geometric information is associated with geometric elements such as points, lines, and faces. Annotation information includes dimension, tolerance, roughness, and other elements related to machining requirements. The dimension information expresses the topological relationship between geometric elements, and the relative position between geometric elements can be determined by the dimension. Tolerance and roughness together express the machining accuracy requirements of parts.

Currently, there are two main expression schemas of 3D models, which are CSG and B-Rep. B-Rep representation mainly expresses geometric and topological information. The former is composed of position, coordinate, curve equation, etc., and the latter is formed by point, edge, contour, surface, and logical relationship among them. Most current modeling software supports B-Rep representation.

3.1. MBD Model Data Structure

Geometric elements are topologically connected to form a solid model. From the perspective of data structure, it is mainly composed of surfaces and edges. Annotation information expresses process information of the model and is the basis for formulating machining route. Most 3D modeling software use surfaces, contours, and edges to describe solid models. Surfaces and edges are the main carriers of model information, and contours, as auxiliary geometric elements, play a transitional role between surfaces and edges. The annotation information of the MBD model mainly includes dimension, tolerance, and surface roughness, which are attached to the model geometry.

The model elements involved in feature recognition mainly include surface, edge, dimension, tolerance, and roughness. Structure arrays are defined to represent model elements, as shown in Table 1. Among them, the arrays surface [], edge [], contour [], surfinish [], gtol [], and dimension [], respectively store the information on surfaces, edges, contours, tolerances, and dimensions of the model.

Table 1. Structure arrays for the storage of MBD model.

Structure Array	Surface [ ]	Edge [ ]	Contour [ ]	Surfinish [ ]	Gtol [ ]	Dimension [ ]
Structure member	ID	ID	Type	Symbol	Symbol	Symbol
	Type	Type	Edge ID	Value	Type	Type
	Concavity–convexity	Concavity–convexity	/	Related surface ID	Value	Value
	Adjacent surface ID	Adjacent surface ID	/	/	Related surface ID	Upper deviation
	Edge ID	Inner contour attribute	/	/	Datum ID	Lower deviation
	Contour information	/	/	/	/	Related surface ID
	Roughness	/	/	/	/	/
	Tolerance	/	/	/	/	/

The contour is divided into inner contour and outer contour. The inner contour is a closed loop formed by the end-to-end connected edges inside the model surface. There can be multiple inner contours on a surface. The outer contour is a closed loop formed by the end-to-end connected edges on the outermost side of the model surface. There is only one outer contour on a surface. Each edge has two or more adjacent faces, and the inner contour properties of some edges on different faces are different. It is specified that if an edge appears as an inner contour edge on an adjacent face, the edge is an inner contour edge.

3.2. Extraction of MBD Model Information

The geometry and topology information and annotation information contained in the MBD model are the data basis for subsequent machining feature recognition and process route optimization. There are mainly two methods to extract model data. The first method is to extract information from neutral format files [44], and the other method is through customized development tools provided by commercial CAD modeling software [45]. In this paper, the latter method is used to develop an automatic model data extraction system and construct the data foundation of a machining feature recognition system based on MBD. This paper will take Creo software as an example to extract model information through Creo Toolkit.

The geometry and topology information of the model includes geometric information and topology information. Among them, geometric information describes the geometric data of vertices, edges, and faces; topology information describes the connection between vertices, edges, and faces. According to the above description of model information, when extracting geometric information, the extracted surface data includes ID and type of the surface and the contour information on it. The extracted edge data is the ID and type of the edge. The extracted topological information is mainly the connection relationship between model surfaces and edges, including the concavity–convexity of surfaces and edges and their adjacent surfaces.

When extracting model information, first use the Creo Toolkit interface function to access the model. Then traverse the model surfaces, and record their ID, type, concavity–convexity of the surface, and the adjacent surface ID in sequence. Traverse the contours on the surface, judge the type of the contours and record its constituent edges. Finally, traverse the edges of the model, and record its ID, type, concavity–convexity, and the ID of the adjacent surface in turn.

Take the Creo model as an example, and build its geometry and topology information extraction flow chart, as shown in Figure 1. The main functions related to model geometry and topology information extraction in Creo Toolkit are shown in Table 2.

The pseudo-code of the geometric and topology information extraction algorithm is shown in Appendix A Algorithm A1.

The extracted model annotation information mainly includes dimension, tolerance, and roughness. Dimension information includes dimension symbols, types, values, deviations, and IDs of associated geometric elements. Tolerance information includes tolerance symbols, types, values, IDS of associated geometric elements, and datums; The surface roughness includes roughness value and associated surface ID. The above information is obtained by traversing the surface dimensions, tolerances, and roughness.

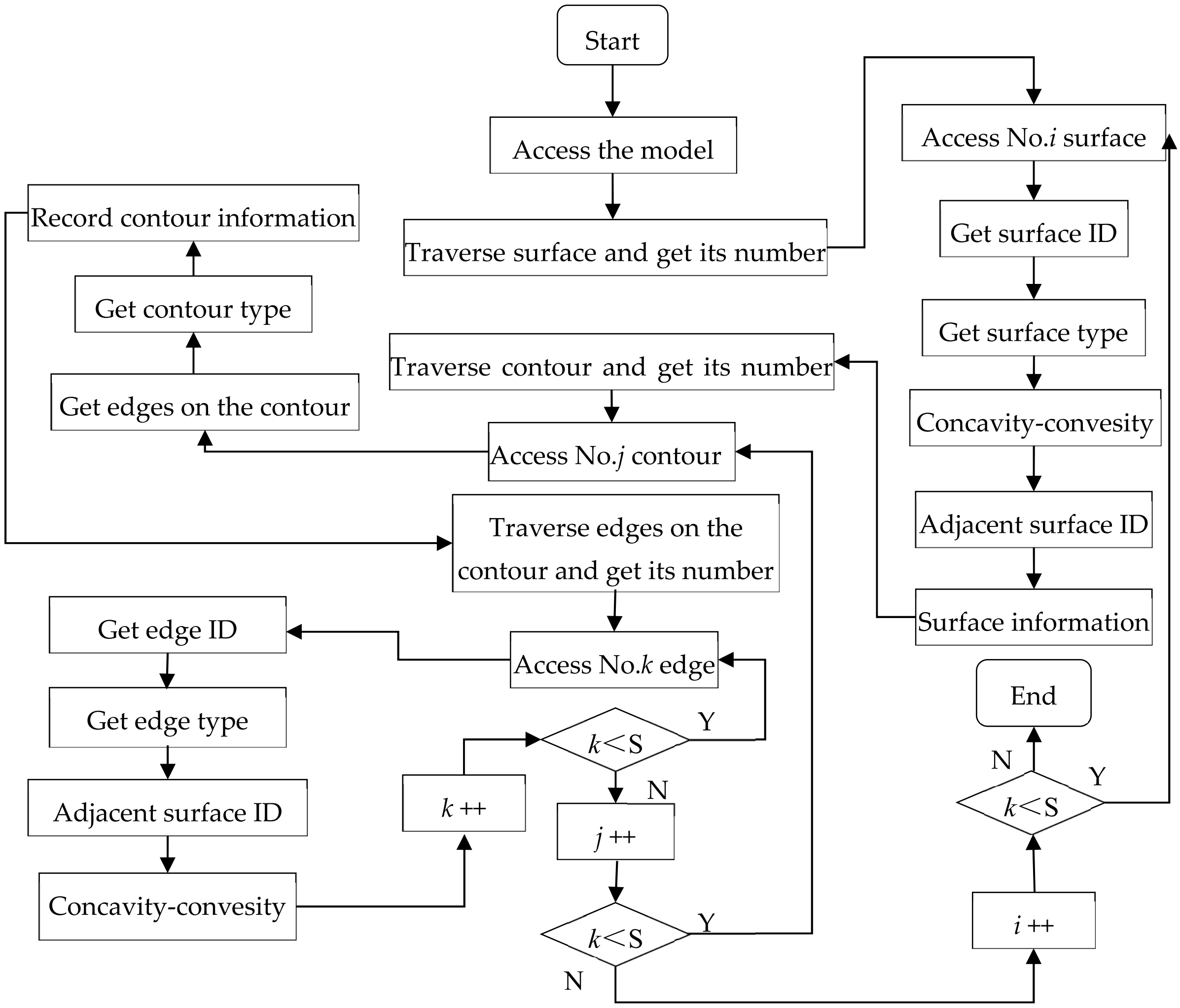


Figure 1. Flow chart of model geometric topology information extraction.

Table 2. Functions of geometric and topology information extraction.

No.	Name	Function
1	ProMdlCurrentGet	Get the handle of current model
2	ProSolidSurfaceVisit	Traverse the model surfaces
3	ProSurfaceContourVisit	Traverse the contour on the surface
4	ProContourEdgeVisit	Traverse the edge on the contour
5	ProContourTraversalGet	Get the type of contour
6	ProSurfaceIdGet	Get the surface ID
7	ProSurfaceTypeGet	Get the type of the surface
8	ProSurfaceDataGet	Get the surface data
9	ProEdgeIdGet	Get the edge ID
10	ProEdgeTypeGet	Get the edge type

4. Machining Feature Recognition Based on AAM

Based on the preceding MBD model information extraction, a machining feature recognition algorithm based on Attributed Adjacency Matrix (AAM) is proposed. The model AAG is constructed by extracting the model information and then decomposed into feature subgraphs and converted into AAM, which is matched with the predefined features to realize the recognition of machining features.

In the process of machining feature recognition based on AAM, a model AAG is constructed according to the geometric and topology information of the model, and it is decomposed based on Decomposition Base Surface (DBS) and Joint Base Surface (JBS). Then match the decomposed AAG subgraph with predefined features to identify machining features. The process is shown in Figure 2.

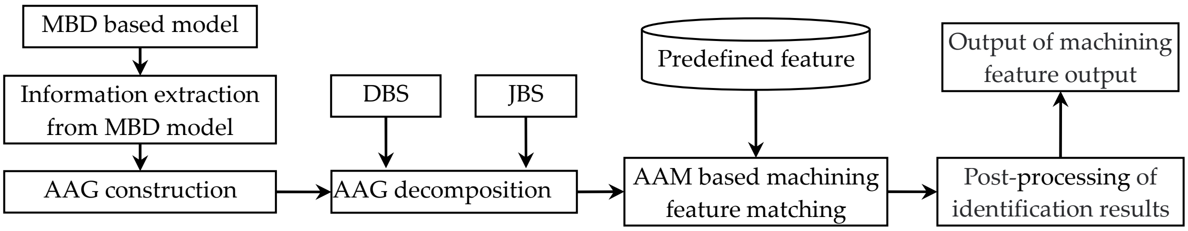

MBD based model
Information extraction from MBD model
AAG construction
DBS
JBS
AAG decomposition
Predefined feature
AAM based machining feature matching
Post-processing of identification results
Output of machining feature output

Figure 2. Flow chart of machining feature recognition.

The related concepts involved in the research process are as follows.

DBS [46] refers to the surface containing one or more inner contours in a 3D model. As shown in Figure 3, the top surface f of the hexahedron contains an inner contour composed of four model edges, and the surface f is a DBS.

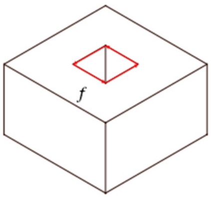


Figure 3. DBS.

Feature Base Surface (FBS) refers to a concave plane in a 3D model, and there is no entity of the model along the normal direction of the surface. As shown in Figure 4, the bottom plane of the keyway is an FBS, which belongs to the concave plane, and there is no entity of the model along its normal direction.

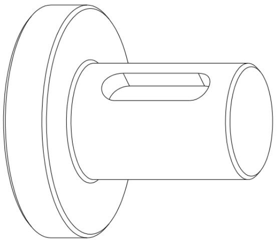


Figure 4. FBS.

JBS refers to two unconnected model surfaces, which need to be processed as the same surface when machining. When two surfaces have the same equation expression, and at least one of them is concave, the two surfaces are JBSs, such as f_3 and f_4, in Figure 5.

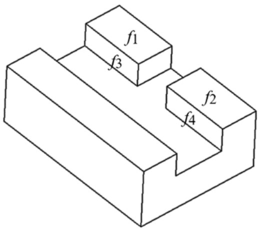


Figure 5. Joint base surface.

The Maximum Concave Connection Subgraph (MCCS) [18] of a node is the maximum connected graph that can be obtained by taking this node as the starting node and extending outward along the node line of the concave attribute in an AAG.

4.1. Construction of Model AAG

In the process of machining feature recognition based on AAG, the B-Rep information of the model is expressed by AAG, which can accurately describe the topological structure of the model surfaces and edges. Conventional AAG uses nodes to represent surfaces, and lines represent the intersecting edges of two surfaces. When the intersecting edge is convex, it is represented by the number 1, and when it is concave, it is represented by the number 0. This representation that does not include other attributes has certain limitations. Sometimes, AAG cannot accurately express the model.

When constructing the model AAG, the attributes of model surfaces and edges are extracted and used to enhance the expression ability of AAG. The node of AAG represents the model surface, and its attributes include type, concavity–convexity, DBS attribute, FBS attribute, and JBS attribute; its definition is shown in Table 3. The line between AAG nodes represents model edges, and its attributes include type, concavity–convexity, and inner contour edges; its definition is shown in Table 4.

Table 3. Attribute of AAG nodes.

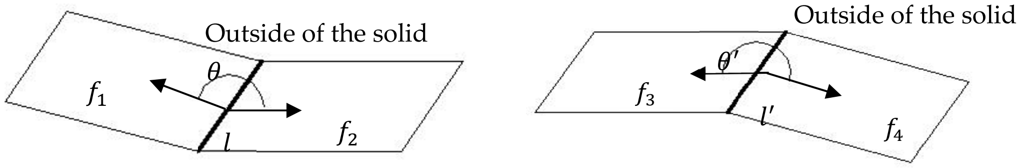
Attribute Name	Type of Variable	Attribute Definition
Type	int	$\begin{cases} 1 & \text{Planar Plane} \\ 2 & \text{Cylinder} \\ 3 & \text{Conical surface} \\ 4 & \text{other surface} \end{cases}$
Concavity–convexity	int	$\begin{cases} -1 & \text{Concave surface} \\ 1 & \text{Convex surface} \end{cases}$
DBS attribute	bool	$\begin{cases} \text{true} & \text{DBS} \\ \text{false} & \text{Not DBS} \end{cases}$
FBS attribute	bool	$\begin{cases} \text{true} & \text{FBS} \\ \text{false} & \text{Not FBS} \end{cases}$
JBS attribute	int	/

Table 4. Attribute definition of node connection in AAG.

Attribute Name	Type of Variable	Attribute Definition
Type	int	1 Straight edge 2 Arc edge 3 Other edge
Concavity–convexity	int	−1 Concave edge 0 Smooth edge 1 Convex edge
Inner contour	bool	true Inner contour edge false No inner contour edge

4.1.1. Concavity–Convexity Judgment of Model Edge

The concavity–convexity of model edges is an important attribute of AAG, which determines the properties of model features. Zhou defined edge concavity–convexity as two adjacent surfaces on the mode; when the angle outside the solid is less than 180°, the intersection line is a concave edge. When the angle outside the solid is greater than 180°, the intersection line is convex [47]. According to the above definition, the edge *l* in Figure 6a is concave, and *l*′ in Figure 6b is convex.



(**a**) (**b**)

Figure 6. The definition of edge concavity–convexity. (**a**) *l* is concave; (**b**) *l*′ is convex.

The preceding method defines the concavity–convexity of the intersection line when the plane intersects with the plane, but does not involve the concavity–convexity judgment of the intersection edge between the surface and planar plane or between two surfaces. According to the extraction of model information in this paper, we propose the definition of model edge concavity–convexity is as follows: when the adjacent surface of the edge is a plane, and the included angle of the non-solid side of the model is in the interval (π, 2π), this edge is convex. When the included angle is in the interval (0, π), this side is concave. When the adjacent surfaces are not all planar planes, and the included angle between the tangent plane of the surface at one point and another plane (or the tangent plane of the surface) on the non-solid side is in the interval (0, π), this edge is concave. When the included angle is in the interval (π, 2π), this side is convex, and when the included angle is equal to 180°, this side is smooth, as shown in Figure 7.

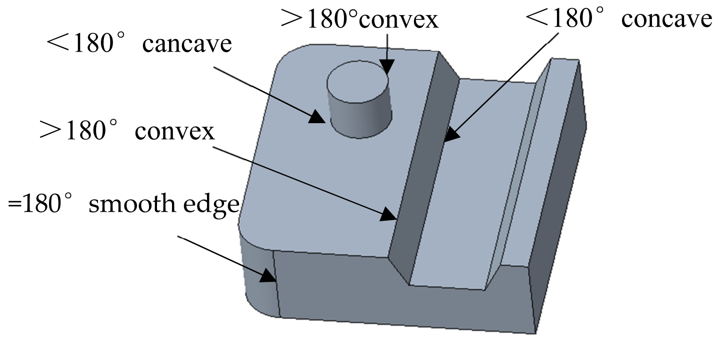


Figure 7. Examples of concavity and convexity of edges in the model.

The concavity–convexity of an edge is determined by the included angle of two planes where the edge is located. Luo [21] proposed a method to solve the included angle by using the arccosine of the plane normal vector, as shown in Formula (1).

$$\theta = arccos\frac{\boldsymbol{n}_1 \cdot \boldsymbol{n}_2}{|\boldsymbol{n}_1||\boldsymbol{n}_2|} \tag{1}$$

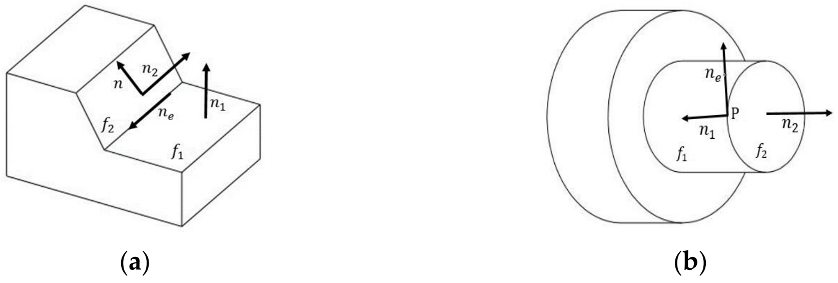
where, θ is the angle between two planes, $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ is the normal vector of the plane.

The included angle that can be calculated in Formula (1) ranges from 0 to π, which can only be applied to the calculation of concave edges. A method to define the included angle between two surfaces is proposed, which can be used to solve the included angle of any angle between two planes or surfaces. The algorithm is as follows:

(1) When the adjacent surfaces of the edges are all planar planes, select one as the base plane, define the edge direction vector as the vector product of the base plane normal vector and the non-base plane normal vector, define the reference direction vector as the vector product of the edge direction vector and the non-base plane normal vector, and the angle between the reference direction vector and the base plane normal vector is θ. When $\cos(\theta) > 0$, the edge is concave. When $\cos(\theta) < 0$, the edge is convex.

(2) When the adjacent surfaces of the edge are not all planar planes, first, take a point P on the edge. If the adjacent surfaces are planes, then select this plane as the base plane, and the tangent plane of the surface at point P is the non-base plane. If the adjacent surfaces are all curved surfaces, the tangent plane of any curved surface at point P is taken as the base surface, and the tangent plane of the other curved surface is taken as the non-base surface. It is defined that the tangent vector at point P is the vector product of the normal vector of the base plane and the normal vector of the non-base plane, the reference direction vector is the vector product of the tangent vector at point P and the normal vector of the non-base plane, and the angle between the reference direction vector and the normal vector of the base plane is θ. When $\cos(\theta) > 0$, the edge is concave. When $\cos(\theta) < 0$, the edge is convex. When $\cos(\theta) = 0$, the edge is smooth.

Take Figure 8 as an example to judge the concavity–convexity of the edges. In (a), f_1 and f_2 are planar planes, and their normal vectors are $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$. Take f_1 as the base plane, the direction vector of the straight side e is $\boldsymbol{n}_e = \boldsymbol{n}_1 \times \boldsymbol{n}_2$, and the reference direction vector is $\boldsymbol{n} = \boldsymbol{n}_e \times \boldsymbol{n}_2$. The included angle of $\boldsymbol{n}$ and $\boldsymbol{n}_1$ is θ. Because of $cos\theta > 0$, the edge is concave. In (b), f_1 is a surface, f_2 is planar, f_2 is taken as the base plane, and then take a point P on the intersection of two surfaces. $\boldsymbol{n}_1$ is the normal vector of the tangent plane of f_1 at point P, $\boldsymbol{n}_2$ is the normal vector of f_2, the tangent vector of the intersection at point P is $\boldsymbol{n}_e = \boldsymbol{n}_2 \times \boldsymbol{n}_1$, and the reference direction vector is $\boldsymbol{n} = \boldsymbol{n}_e \times \boldsymbol{n}_1$. The angle between $\boldsymbol{n}$ and $\boldsymbol{n}_2$ is θ. Because of $\cos(\theta) < 0$, the edge e is convex.



(**a**) (**b**)

Figure 8. Concavity–convexity judgment of model edge. (**a**) shows the intersection line of two planar planes; (**b**) shows the intersection line of planar plane and curved surface.

4.1.2. Concavity–Convexity Judgment of Model Surface

Guo's definition of surface concavity–convexity is as follows [48]. If all the edges of the outer contour are convex, the face is convex, and if not, it is concave. But in Figure 9, the outer contour edges of f_1 and f_2 contain concave edges, and according to the above definition, both surfaces are concave. However, it is generally considered that f_1 is convex. So, the above definition is not accurate enough to express the topological information of the model and is only applicable to the model without convex features.

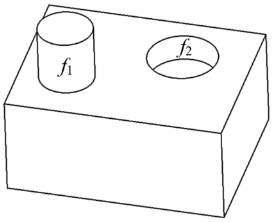


Figure 9. Concavity–convexity judgment of model surface.

A new method to define the surface convexity–concavity is proposed. The rules are as follows: (1) For a planar plane, when all its outer contour edges are convex, the surface is convex, and when the outer contour edges of the surface are not all convex, the surface is concave. (2) For a quadric surface, its concavity–convexity is judged according to whether the normal vector at any position on the surface is on the same side as the curvature circle at that point. When it is on the same side, the surface is concave; otherwise, it is convex. This is shown in Figure 10.

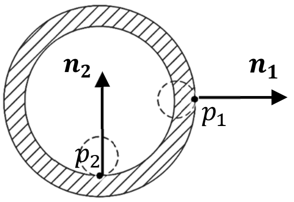


Figure 10. Judgment of curvature of surface.

4.1.3. Construction of AAG

Based on the preceding model information extraction method and attribute research, the model AAG is constructed by taking the model surface as the node, taking the intersection line as the edge, and adding the attributes of surface and line. Its process is shown in Figure 11, and the main steps are as follows.

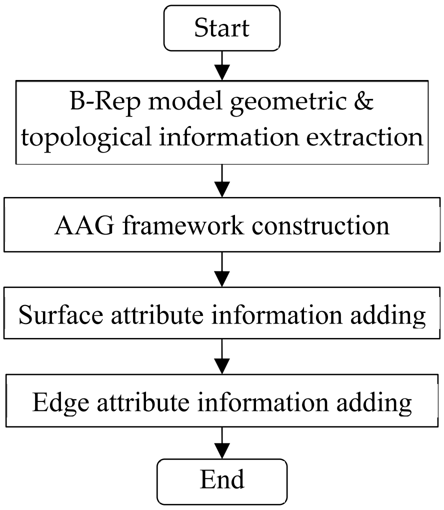

Start
B-Rep model geometric & topological information extraction
AAG framework construction
Surface attribute information adding
Edge attribute information adding
End

Figure 11. AAG construction flow.

(1) Extract B-Rep model information. Taking Creo models as examples, the Creo Toolkit is used to traverse the model surfaces and lines to extract their attribute.

(2) Model AAG framework is established, which is based on the geometric information and topological relationship between surface and line. AAG nodes are created based on model surfaces, and AAG edges are created based on the topological relationship between surfaces and edges.

(3) Add model surface attribute information. Traverse AAG nodes and add their attribute information.

(4) Add model edge attribute information. Traverse AAG edges and add their attribute information.

Based on the preceding research on model information extraction and AAG construction, the serialization of model AAG is realized by building structure arrays and assigning values to the structure members. The main steps are as follows:

(1) Access the structure arrays surface [], edge [], and contour [], which were assigned in the previous information extraction.

(2) Construct node structure array node [] and line structure array line [], and establish the corresponding relationship between surface [] and node [] and between edge [] and line [].

(3) Traverse and extract the data in surface [], and define the relevant attributes of node []. The data in edge [] is extracted to define the relevant attributes of the node connection line [].

After the above operations, the model AAG is serialized in the form of a structure array, and its pseudo-code is shown in Appendix A Algorithm A2.

4.2. AAG Decomposition and Reconstruction Based on DBS and JBS

The essence of machining feature recognition based on AAG is to search for isomorphic subgraphs matching predefined features in model AAG [49]. Graph matching is a typical NP-complete problem. So far, there is no suitable method to solve it. The usual countermeasure is to decompose the AAG and then match the subgraphs obtained after decomposition with predefined features. In most of the previous studies, AAGs are decomposed according to the concavity–convexity of the model edges, which has high accuracy for concave feature recognition. However, for convex features, it is not ideal.

An AAG decomposition and reconstruction algorithm based on DBS and JBS is proposed. According to the node attributes, AAG is decomposed to realize the recognition of various features, and the features with the same JBS are reconstructed into subgraphs to merge the segmented features. The basic principle is as follows. AAG is searched for nodes with DBS attributes, and according to the concavity–convexity attributes of the inner contour edges, the subgraphs with concave and convex features are extracted respectively. It is further judged whether the convex feature subgraphs still contain concave feature subgraphs, and the nodes corresponding to the non-machining surfaces in the remaining AAG are deleted. The MCCS with the FBS attribute is extracted. Finally, the nodes with JBS attributes are searched and merged to realize the subgraph reconstruction within the model. The main steps are as follows.

(1) AAG decomposes based on DBS. Traverse AAG to find DBS in the model and delete the inner contour on it. If the edge of the inner contour is a convex edge, the decomposed subgraph is merged into the concave feature subgraph set MAAG. Otherwise, the subgraph is merged into the convex feature subgraph set TAAG.

(2) If a concave edge exists in the convex feature subgraph TAAG-*i*, the MCCS CAAG-*i* is extracted with the adjacent face of the edge as the starting node, and CAAG-*i* and the remaining graphs are merged into the feature subgraph set MAAG. Otherwise, TAAG-*i* is set as an independent feature and is merged into the feature subgraph set MAAG.

(3) Delete the surface of the non-machining feature. AAG is updated to AAG-1, and the node with the convex surface attribute in AGG-1 and the edges connected with this node are deleted.

(4) AAG decomposes based on FBS. AAG-1 is updated to AAG-2, the nodes with FBS attribute are taken as start nodes, and the MCCS is extracted, which is put into the feature subgraph set MAAG.

(5) Subgraph reconstructs based on JBS. Search for nodes with the same JBS attribute in the feature subgraph set MAAG and merge them to obtain the reconstructed feature subgraph.

The pseudo-code of the preceding algorithm is shown in Appendix A Algorithm A3.

Taking the model shown in Figure 12 as an example, the process of AAG decomposition and reconstruction based on DBS and JBS will be described. According to the method in Section 4.2, its AAG is constructed as shown in Figure 13. There are 16 nodes and 35 lines in the graph, which represent the number of model surfaces and edges, respectively. The numbers on the line represent the edge concavity–convexity, "1" represents the convex edge, and "−1" represents the concave edge. The red lines represent the inner contour edge attribute, the red nodes represent FBS attribute, the orange nodes represent DBS attribute, and the green nodes represent JBS attribute.

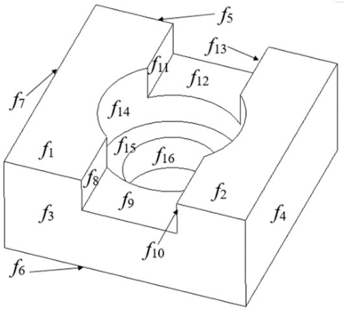


Figure 12. MBD model. Where, f_1–f_{16} represent the 16 surfaces of the model.

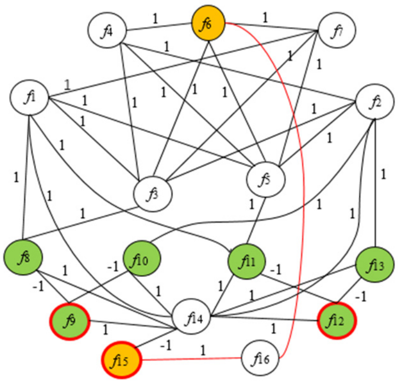


Figure 13. Model AAG.

The AAG decomposition process of the above model is shown in Figure 14. First, search for the nodes with DBS attribute, which are f_{15} and f_6. Search for lines that are connected to f_{15} and f_6 and have inner contour attributes, which are the two red lines in the AAG, and delete them. Since f_6 has convex edge attribute, the subgraph (I) is composed of a single node f_6, which is a concave characteristic subgraph. Then, update AAG to get AAG-1, as shown in (a). Delete the node with the convex attribute in AAG-1 and the line connected with the node, and update to get AAG-2, as shown in (b). Search AAG-2 for nodes with FBS attribute, which are f_9, f_{12}, and f_{15}. Finally, the MCCSs of nodes f_9, f_{12}, and f_{15} are extracted, respectively, to obtain the feature subgraph (II) composed of nodes f_8, f_9, and f_{10}; the feature subgraph (III) composed of nodes f_{11}, f_{12}, and f_{13}; and the feature subgraph (IV) composed of nodes f_{14} and f_{15}. After decomposition, a total of four feature subgraphs are obtained, as shown in (c), which are the through-hole feature where f_{16} is located, the blind-hole feature where f_{15} is located, the groove feature where f_9 is located, and the groove feature where f_2 is located.

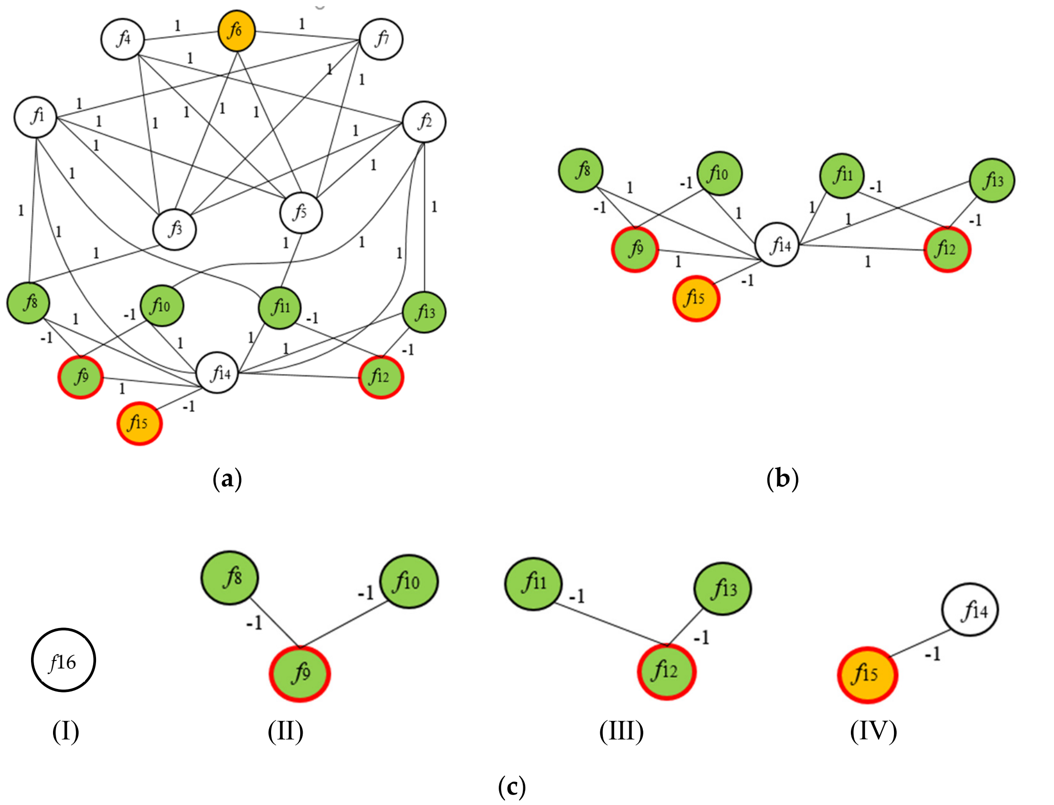


Figure 14. AAG decomposition based on DBS. (**a**) shows the updated AAG-1; (**b**) shows the updated AAG-2; (**c**) shows the four decomposed subgraphs.

Surface f_9 and f_{12} are JBSs, and they will generally be processed as a feature when machining. However, it is now divided due to the intersection of features, so two feature subgraphs are generated through the decomposition of AAG. According to the AAG reconstruction method based on JBS described above, it is found that three groups of nodes, f_8 and f_{11}, f_9 and f_{12}, and f_{10} and f_{13}, have the same JBS attribute. Through subgraph reconstruction, the two features corresponding to nodes f_8, f_9, f_{10} and f_{11}, f_{12}, f_{13} are merged into one through slot feature, as shown in Figure 15.

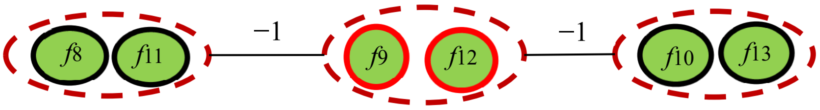


Figure 15. Subgraph reconstruction based on JBS.

4.3. Machining Feature Matching Based on AAM

Generally, feature matching is accomplished by the isomorphism of graphs. Its principle is as follows: after the decomposition of AAG, the feature subgraph is isomorphically matched with the predefined features. If the two have the same topology, it indicates that the feature matching is successful, and the feature object is recognized as the corresponding processing feature; if the matching fails, the feature type is undefined or does not exist. In order to overcome the shortcomings of the existing feature matching methods, a machining feature matching method based on AAM is proposed.

4.3.1. Machining Feature Matching Method Based on AAM

Aiming at the problem of feature matching, a feature matching algorithm based on AAM is proposed. The decomposed feature subgraph is constructed as AAM and matched with the predefined feature to realize the recognition of machining features. The advantage of this method is that it only needs to match the features of the constituent elements of AAM and avoids the similarity transformation of AAM.

The AAM constructed by AAG is a square matrix whose order is equal to the number of nodes in the feature subgraph. Diagonal elements represent the attributes of nodes in the feature subgraph, and non-diagonal elements represent the attributes of edges in the feature subgraph. The AAM of a feature subgraph is shown in Formula (2).

$$M[n][n] = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix} \tag{2}$$

where, n represents the number of nodes of the feature subgraph, m_{ij} $(i \neq j)$ indicates the attribute of edge, m_{ii} indicates the attribute of the node.

The node attribute is composed of concavity–convexity, type, and FBS attribute of the surface. It is expressed as a positive integer containing three digits. It is recorded as $m_{ii} = abc$, and the values of each attribute are shown in Formulas (3)–(5).

$$a = \begin{cases} 1 & \text{Convexity surface} \\ 0 & \text{Concavity surface} \end{cases} \tag{3}$$

$$b = \begin{cases} 1 & \text{Planar plane} \\ 2 & \text{Cylinder} \\ 3 & \text{conical surface} \\ 4 & \text{other surface} \end{cases} \tag{4}$$

$$c = \begin{cases} 1 & \text{FBS} \\ 2 & \text{Non FBS} \end{cases} \tag{5}$$

The edge attribute consists of concavity–convexity and the type of the edge, which is represented by a positive integer containing two digits and is recorded as $m_{ij} = de$. The values of each attribute are shown in Formulas (6) and (7).

$$d = \begin{cases} 1 & \text{Concavity edge} \\ 2 & \text{Convexity edge} \\ 3 & \text{smooth edge} \end{cases} \tag{6}$$

$$e = \begin{cases} 1 & \text{Staight edge} \\ 2 & \text{Arc edge} \\ 3 & \text{other edge} \end{cases} \tag{7}$$

The node and edge attributes are stored in the structure array node [][] and line [][], respectively. Node attribute can be assessed through node [ ][ ].node_at, node[ ][ ].node_ type and node[ ][ ].node_ f_face. Edge attribute can be obtained through line[ ][ ].line_ at and line[ ][ ].line_type.

If the geometric information and topological structure of the nodes and edges in the predefined feature and feature subgraph are the same, it means that they match, which should meet the requirement of the same order of the matrix, same diagonal elements, and the similarity of each row element set. The similarity of each row element set means that each row element of the AAM of the feature subgraph has a one-to-one mapping relationship with each row element of the predefined AAM.

The main process of the AAM-based feature matching algorithm is as follows.

(1) Judge whether the order of the feature subgraph AAM and the predefined AAM are the same.

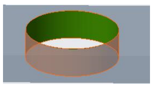
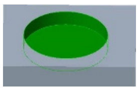
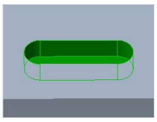
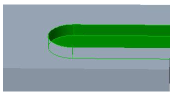
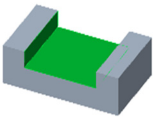
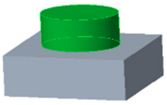
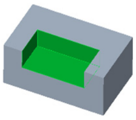
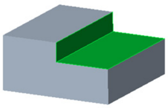
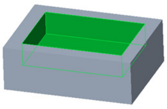
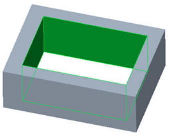
(2) Judge whether diagonal elements are equal.

(3) Judge whether the sets formed by each row element are similar. If any of the conditions are not met, the matching is unsuccessful.

(4) If the matching is not successful, the sub-feature is exported, and whether they are machining features will be judged by human–computer interaction. If so, they will be added to the predefined feature library.

Table 5 shows typical machining features and their predefined AAMs, in which the green area is the machining feature. The algorithm in this paper can effectively identify through hole, blind-hole, closed key slot, open key slot, through groove, cylinder extrusion, blind groove, step, and other machining features.

Table 5. Typical machining features.

No.	Feature Name	Graphical Representation	AAM
1	Through-hole		$\begin{bmatrix}220\end{bmatrix}$
2	Blind-hole		$\begin{bmatrix}220 & 21\\ 21 & 110\end{bmatrix}$
3	Closed key slot		$\begin{bmatrix}220 & 13 & 0 & 13 & 21\\ 13 & 210 & 13 & 0 & 11\\ 0 & 13 & 220 & 13 & 21\\ 13 & 0 & 13 & 210 & 11\\ 21 & 11 & 21 & 11 & 110\end{bmatrix}$
4	Open key slot		$\begin{bmatrix}110 & 11 & 21 & 11\\ 11 & 210 & 13 & 0\\ 21 & 13 & 220 & 21\\ 11 & 0 & 21 & 210\end{bmatrix}$
5	Through groove		$\begin{bmatrix}210 & 11 & 0\\ 11 & 110 & 11\\ 0 & 11 & 210\end{bmatrix}$
6	Cylinder extrusion		$\begin{bmatrix}111 & 22\\ 22 & 221\end{bmatrix}$
7	Blind groove		$\begin{bmatrix}210 & 11 & 0 & 11\\ 11 & 110 & 11 & 11\\ 0 & 11 & 210 & 11\\ 11 & 11 & 11 & 110\end{bmatrix}$
8	Step		$\begin{bmatrix}210 & 11\\ 11 & 210\end{bmatrix}$
9	Rectangular blind cavity		$\begin{bmatrix}210 & 11 & 0 & 11 & 11\\ 11 & 210 & 11 & 0 & 11\\ 0 & 11 & 210 & 11 & 11\\ 11 & 0 & 11 & 210 & 11\\ 11 & 11 & 11 & 11 & 110\end{bmatrix}$
10	Rectangular through cavity		$\begin{bmatrix}210 & 11 & 0 & 11\\ 11 & 210 & 11 & 0\\ 0 & 11 & 210 & 11\\ 11 & 0 & 11 & 210\end{bmatrix}$

4.3.2. AAM Establishment Algorithm

For a characteristic subgraph t, the structure arrays of the nodes and edges are node[t][n] and line[t][n]. Traverse the members of the node array node[ ][ ].node_at, node[ ][ ].node_ type and node[ ][ ].node_ f_face and get the node properties. Traverse the edge array members line[ ][ ].line_ at and line[ ][ ].line_type and get the edge properties. The establishment process of AAM is shown in Figure 16.

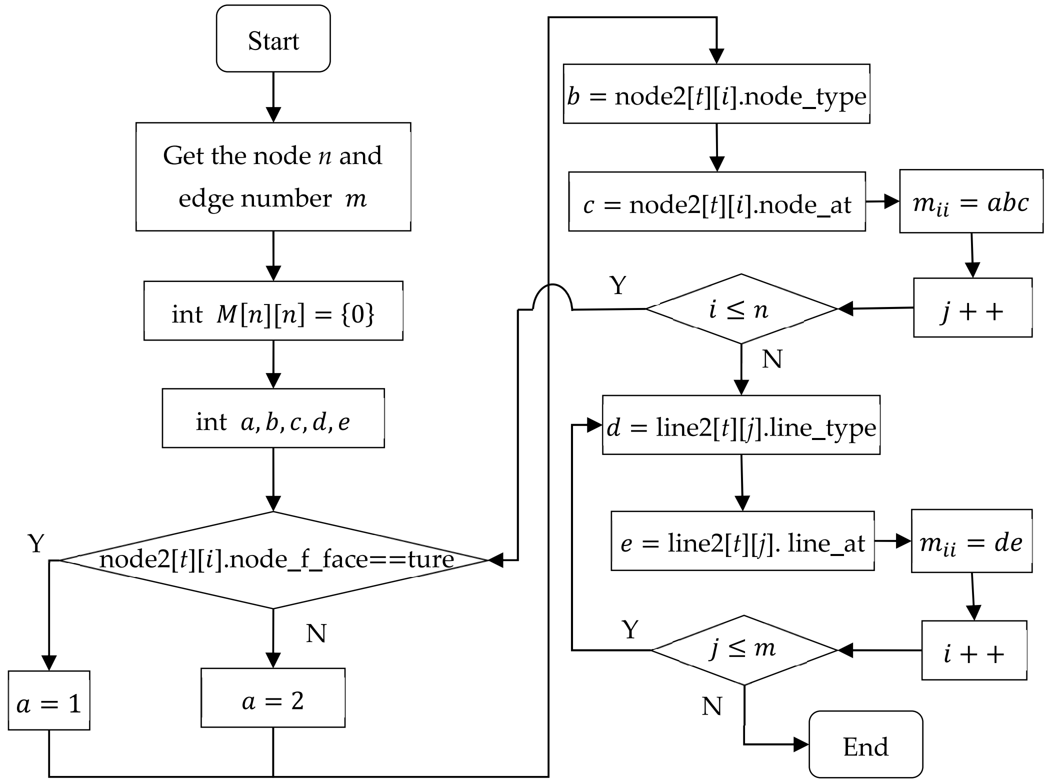

Start
Get the node n and edge number m
int $M[n][n] = \{0\}$
int a, b, c, d, e
node2[t][i].node_f_face==ture
Y
N
$a = 1$
$a = 2$
b = node2[t][i].node_type
c = node2[t][i].node_at
$m_{ii} = abc$
$j++$
Y
$i \le n$
N
d = line2[t][j].line_type
e = line2[t][j]. line_at
$m_{ii} = de$
Y
$j \le m$
$i++$
N
End

Figure 16. The AAM construction process.

The main steps are as follows, and the pseudo-code of the algorithm is shown in Appendix A Algorithm A4.

(1) Access the data of line2[t][] and node2[t][], and define matrix $M[n][n]$, and assign the initial value of each element to 0.

(2) Determine the value of m_{ii}. If node2[t][i].node_f_face = true, then $a = 1$. If node2[t][i].node_f_face = false, then a = 2. Determine the value of b as follows: $b = \text{node2}[t][i]$.node_ type. Determine the value of c as follows: $c = \text{node2}[t][i]$ .node_at.

(3) Assign the data of m_{ij} as follows: $d = \text{line2}[t][j]$ .line_type $ande$ = line2[t][j].line_ at .

4.3.3. AAM Matching Algorithm

A matching method of feature subgraph AAM and predefined feature AAM is proposed. The feature subgraph AAM to be matched is represented by matrix A$[n][n]$, and the predefined feature AAM is represented by matrix B$[m][m]$. The main process of matching between them is as follows.

(1) If $n = m$, go on; otherwise, the matching is unsuccessful.

(2) Define set S_1 and S_2, and assign the values of diagonal elements of A$[n][n]$ to S_1, and the value of B$[m][m]$ diagonal element to S_2. If S_1 = S_2, go on; otherwise, the matching is not successful.

(3) Define set S_{Ai} and S_{Bj} and assign the valve of No.i row of a [n] [n] to set S_{Ai}, the valve of No.j row of a [n] [n] to set S_{Bj}. If each set S_{Ai} can find the set S_{Bj} that is equal to it and does not repeat, the matching is successful and the machining feature is recognized; otherwise, the matching is unsuccessful.

The flow chart of feature subgraph matching is shown in Figure 17, and the pseudo-code of the algorithm is shown in Appendix A Algorithm A5.

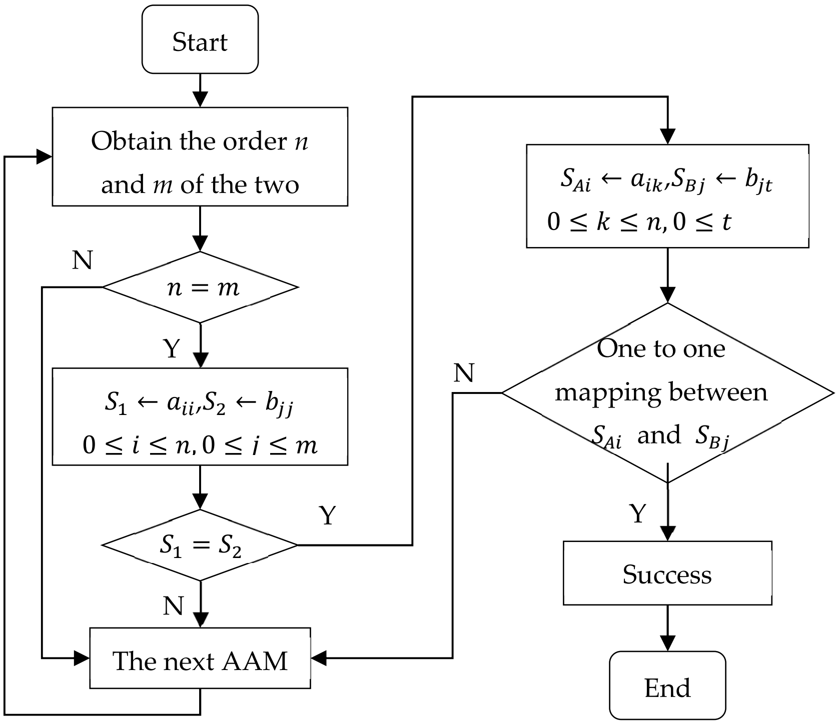

Start
Obtain the order n and m of the two
$n = m$
N
Y
$S_1 \leftarrow a_{ii}, S_2 \leftarrow b_{jj}$
$0 \le i \le n, 0 \le j \le m$
$S_1 = S_2$
Y
N
The next AAM
$S_{Ai} \leftarrow a_{ik}, S_{Bj} \leftarrow b_{jt}$
$0 \le k \le n, 0 \le t$
One to one mapping between S_{Ai} and S_{Bj}
N
Y
Success
End

Figure 17. AAM matching process.

5. Decision Making of Feature Processing Scheme Based on Fuzzy Comprehensive Evaluation

The determination of the feature processing scheme is a complex decision-making process and also the premise of process route optimization. Based on the recognition of MBD model machining features, a feature processing scheme decision method based on fuzzy comprehensive evaluation is proposed. By analyzing the influencing factors of processing scheme selection, the mapping relationship between the influencing factors and processing schemes is established. A multi-factor comprehensive evaluation model is established according to the single-factor evaluation results. By calculating the comprehensive evaluation index vector, the optimization of feature processing schemes is realized. The processing scheme decision-making process based on fuzzy comprehensive evaluation is shown in Figure 18. The main steps are as follows:

(1) Establish the factor set that affects the decision-making of the processing scheme. The set is shown as follows

$$U = \{u_1, u_2, \cdots, u_i\} \tag{8}$$

where, u_i is the No.i factor affecting the evaluation object.

(2) Determine the weight of each influencing factor. In the evaluation of processing methods, the importance of each influencing factor is different. The weight of the influencing factor u_i is expressed as w_i. Therefore, the weight set of influencing factors is established as shown in the following formula

$$W = \{w_1, w_2, \cdots, w_i\} \tag{9}$$

(3) Establish an alternative set of processing schemes. The alternative set of processing schemes is the set of all possible results of the evaluation object.

(4) Single-factor fuzzy evaluation. The fuzzy comprehensive evaluation matrix R is obtained according to the fuzzy relationship matrix between each influencing factor and the alternative set of the processing scheme.

(5) Calculate the fuzzy comprehensive evaluation index vector $B = W{\cdot}R$. Select the largest evaluation index b_i as the final result.

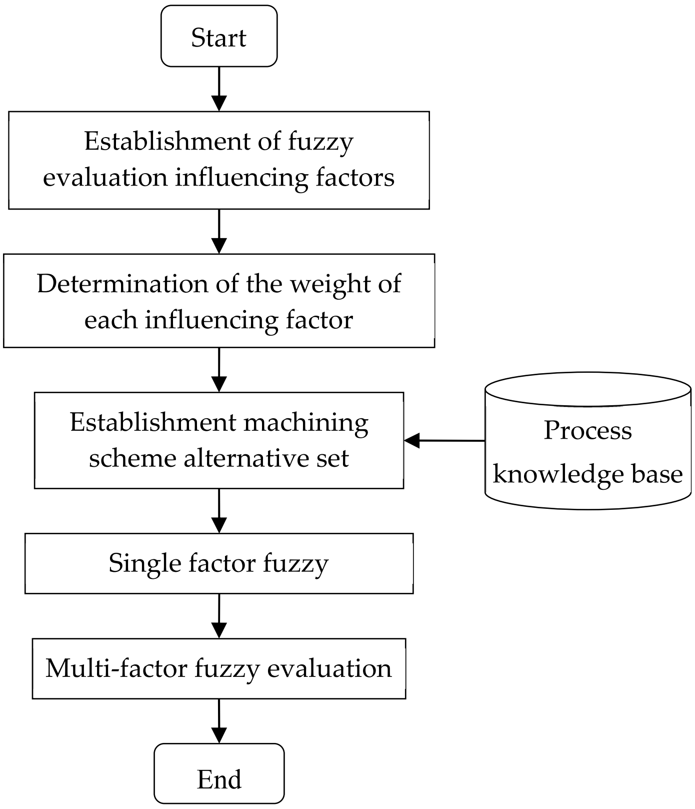

Start
Establishment of fuzzy evaluation influencing factors
Determination of the weight of each influencing factor
Establishment machining scheme alternative set
Process knowledge base
Single factor fuzzy
Multi-factor fuzzy evaluation
End

Figure 18. Decision of feature processing method based on fuzzy comprehensive evaluation.

According to the process characteristics of parts, the features are divided into face features and hole features. We take face features as an example and carry out fuzzy comprehensive evaluation of feature processing schemes on the basis of studying its single factor fuzzy evaluation.

5.1. Establishment of Alternative Sets for Fuzzy Evaluation of Surface Features

The main influencing factors of the surface feature process scheme include workpiece material, feature type, processing accuracy, roughness, and other process requirements. The fuzzy evaluation influencing factor set is shown in the following formula

$$U_1 = \{M, F, IT, R_a\} \tag{10}$$

where, M stands for workpiece material, F stands for feature type, IT stands for machining accuracy and R_a stands for surface roughness. The workpiece material M is specifically expressed as:

$$M = \{m_1, m_2, m_3\} \tag{11}$$

where, $m_1 \sim m_3$ represent steel, cast iron, and non-ferrous metals, respectively.

Feature type F is represented by the set shown in the following formula

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_6\} \tag{12}$$

where, $f_1 \sim f_6$ denote plane, step, blind step, through groove, blind groove, cavity, and boss, respectively.

The alternative set of fuzzy evaluation of feature processing schemes is the set of all possible feature processing schemes. The type, processing size, material, and accuracy requirements of features are the main factors to be considered in formulating processing schemes. The typical processing scheme set of surface processing features is shown in Table 6.

Table 6. Typical processing scheme of surface features.

No.	Processing Scheme	Machining Accuracy	Surface Roughness *Ra*/μm	Scope of Application
1	Rough planing	IT11~13	12.5~50	Plane processing of ferrous and nonferrous metals
2	Rough planing—fine planing	IT8~10	3.2~6.3	
3	Rough planing—fine planing—wide blade fine planing	IT6~7	0.8~1.6	
4	Rough milling	IT11~13	6.3~25	Plane, through groove, step, and other features of ferrous and non-ferrous metals
5	Rough milling—finish milling	IT8~10	1.6~6.3	
6	Rough milling—finish milling—fine grinding	IT6~7	0.8~1.6	Plane of ferrous metal

A fuzzy evaluation candidate set of face features is established according to the above processing scheme, as shown below

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6\} \tag{13}$$

where, $x_1 \sim x_6$ represent typical processing schemes of surface features, which are rough planing, rough planing fine planing, rough planing fine planing wide blade fine planing, rough milling, rough milling fine milling, rough milling fine milling fine grinding.

5.2. Determination of Weight of Each Influencing Factor

In order to ensure the objectivity of the weights of the influencing factors of fuzzy evaluation, the method of combining the Analytic Hierarchy Process (AHP) and expert consultation [50] is used to establish the weight set of the influencing factors of area characteristics, as shown below

$$W = \{w_1, w_2, w_3, w_4\} \tag{14}$$

where, $w_1 \sim w_4$ represent the weight of processing material, feature type, processing accuracy, surface roughness, and $\sum_{i=1}^{4} w_i = 1$.

Because each factor has a different influence on the feature processing scheme, according to the scale of AHP shown in Table 7, a comparison matrix A of surface features is established, as shown in Formula (15).

$$[a_{ij}]_{4\times 4} = \begin{bmatrix} 1 & 2 & 3 & 3 \\ \frac{1}{2} & 1 & 3 & 3 \\ \frac{1}{3} & \frac{1}{3} & 1 & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 & 1 \end{bmatrix} \tag{15}$$

where, $a_{ij} = C_i/C_j, a_{ij} > 0,\ \ a_{ij} = 1/a_{ij}$.

Table 7. Scale table.

Scale Value	Meaning
1	C_i is as important as C_j
3	C_i is slightly more important than C_j
5	C_i is more important than C_j
7	C_i is significantly more important than C_j
9	C_i is extremely more important than C_j
2, 4, 6, 8	Preceding adjacent median
Reciprocal of scale value	Contrary to the above comparison between C_i and C_j

In order to ensure the rationality of the comparison matrix, the consistency test is carried out. The consistency verification ratio CR of the comparison matrix is shown as follows

$$CR = CI/RI \tag{16}$$

where, CI is the consistency index of comparison matrix, which is defined as follows

$$CI = (\lambda - n)/(n-1) \tag{17}$$

where, n is the order of the comparison matrix, and RI is the random consistency index. The values of RI are shown in Table 8.

Table 8. Random consistency index.

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.51

If $CR < 0.1$, it is considered that the consistency of the comparison matrix is satisfactory, and its normalized eigenvector can be used as the weight vector. Through calculation, the maximum eigenvalue of the comparison matrix A of surface features is 5.0606, and the consistency verification ratio of its comparison matrix CR is 0.0224, which is less than 0.1. Therefore, the consistency is satisfactory. Normalize the feature vector of a to obtain the weight set of influencing factors $W = \{0.444, 0.312, 0.122, 0.122\}$.

5.3. Single Factor Fuzzy Mapping of Machining Schemes for Surface Features

For surface feature processing schemes, according to the set of influencing factors for fuzzy evaluation, a fuzzy mapping relationship model with each influencing factor is established, as shown in Figure 19.

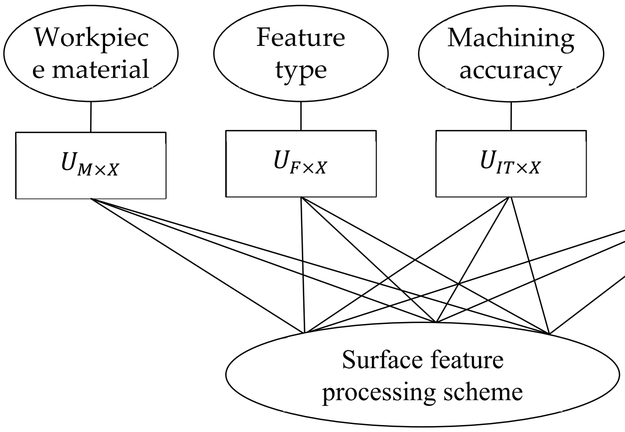


Figure 19. Fuzzy mapping model between machining scheme and influencing factors.

In the preceding figure, $U_{M\times X}$, $U_{F\times X}$, $U_{IT\times X}$, and $U_{Ra\times X}$ are the fuzzy relations between workpiece material, feature type, machining accuracy, surface roughness, and surface feature machining scheme.

5.3.1. Fuzzy Mapping between Workpiece Material and Machining Schemes

According to the application scope of the machining scheme of each surface feature, the corresponding relationship between the workpiece material and the machining scheme is shown in Table 9.

Table 9. Correspondence between workpiece material and surface feature processing scheme.

No.	x_1	x_2	x_3	x_4	x_5	x_6
m_1	√	√	√	√	√	√
m_2	√	√	√	√	√	√
m_3	√	√	√	√	√	

When the workpiece material m_i can be processed by the processing scheme x_j, the membership degree of (m_i, x_j) is set to 1; otherwise, it is set to 0. According to the preceding correspondence, the fuzzy mapping expression between the material type and the surface feature processing scheme is obtained, as shown below

$$f_{M\times X}(m_i, x_j) = \begin{cases} f_{M\times X}(m_1, x_j) = \left\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4}, \frac{1}{x_5}, \frac{1}{x_6}\right\} \\ f_{M\times X}(m_2, x_j) = \left\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4}, \frac{1}{x_5}, \frac{1}{x_6}\right\} \\ f_{M\times X}(m_3, x_j) = \left\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4}, \frac{1}{x_5}, \frac{0}{x_6}\right\} \end{cases} \tag{18}$$

From the above formula, the fuzzy relationship between the workpiece material set M and the surface feature machining scheme set X can be obtained, which is represented by fuzzy matrix $R_{M\times X}$, as shown in Equation (19).

$$[r_{ij}]_{3\times 6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{19}$$

where, $r_{ij} = \mu_{M\times X}(m_i, x_j)$ indicates the possibility of selecting a processing scheme x_j when the workpiece material is m_i.

5.3.2. Fuzzy Mapping between Feature Types and Machining Schemes

The corresponding relationship between the feature types and the processing schemes of the features of each surface is shown in Table 10.

Table 10. Correspondence between feature type and surface feature processing scheme.

No.	x_1	x_2	x_3	x_4	x_5	x_6
f_1	√	√	√	√	√	√
f_2	√	√	√	√	√	√
f_3				√	√	
f_4	√	√	√	√	√	√
f_5				√	√	
f_6				√	√	
f_7				√	√	

When the processing feature of type f_i can be processed with processing scheme x_j, the membership degree of (f_i, x_j) is set to 1; otherwise, it is set to 0. According to the corresponding relationship, the fuzzy mapping expression between the feature type and the surface feature processing scheme is obtained, as shown below.

$$\begin{cases} f_{F\times X}(f_1,x_j)=\left\{\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{1}{x_6}\right\} \\ f_{F\times X}(f_2,x_j)=\left\{\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{1}{x_6}\right\} \\ f_{F\times X}(f_3,x_j)=\left\{\frac{0}{x_1},\frac{0}{x_2},\frac{0}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{0}{x_6}\right\} \\ f_{F\times X}(f_4,x_j)=\left\{\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{1}{x_6}\right\} \\ f_{F\times X}(f_5,x_j)=\left\{\frac{0}{x_1},\frac{0}{x_2},\frac{0}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{0}{x_6}\right\} \\ f_{F\times X}(f_6,x_j)=\left\{\frac{0}{x_1},\frac{0}{x_2},\frac{0}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{0}{x_6}\right\} \\ f_{F\times X}(f_7,x_j)=\left\{\frac{0}{x_1},\frac{0}{x_2},\frac{0}{x_3},\frac{1}{x_4},\frac{1}{x_5},\frac{0}{x_6}\right\} \end{cases} \tag{20}$$

According to Formula (20), the fuzzy relationship between the feature type set F and the surface feature processing scheme set X can be obtained, which is expressed by the fuzzy matrix $R_{F\times X}$, as shown below

$$\left[r_{ij}\right]_{7\times 6}=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{21}$$

where, $r_{ij}=\mu_{F\times X}(f_i,x_j)$, which indicates the possibility of selecting machining scheme x_j when the machining feature type is f_i.

5.3.3. Fuzzy Mapping between Machining Accuracy and Machining Schemes

The machining accuracy range of feature machining scheme x_j is $ITn\sim ITm, n<m$. The following assumptions are made in this paper:

(1) When IT is less than $ITn-2$, the scheme x_j cannot meet the processing requirements, so the possibility of scheme selection is 0.

(2) When IT is greater than ITm, although this scheme can meet the processing requirements, its economy is poor, so the possibility of selecting this scheme is 0.

(3) When IT is in the range of $ITn-2\sim ITn$, the possibility of selecting this scheme is 0.8~1.

(4) When IT is in the range of $ITn\sim ITm$, the possibility of selecting this scheme is 1.

The membership function of the fuzzy mapping between the machining accuracy and the scheme is defined using the piecewise function method, as shown in Formula (22).

$$\mu(IT,x_j)=\begin{cases} 0 & IT<ITn-2 \\ 1-\frac{(ITn-IT)}{5(ITm-ITn)} & ITn-2\le IT<ITn \\ 1 & ITn\le IT\le ITm \\ 0 & IT>ITm \end{cases} \tag{22}$$

According to $\mu(IT,x_j)$, the fuzzy relationship between the machining accuracy IT and the surface feature machining scheme set X is obtained, which is expressed by the fuzzy matrix $R_{IT\times X}$, as shown below

$$R_{IT\times X}=\left[r_{1j}\right]_{1\times 6} \tag{23}$$

where, r_{1j} represents the possibility of selecting the machining scheme x_j when the machining feature accuracy requirement is IT.

5.3.4. Fuzzy Mapping between Surface Roughness and Machining Scheme

When the surface roughness is greater than the upper limit of the roughness obtained by the machining scheme, the machining scheme cannot meet the requirement, and the selectivity of the machining scheme is 0. When the surface roughness is between the upper and lower limits of the scheme roughness, the processing scheme can meet the requirements, and the optional range of the scheme is 0.8~1. When the surface roughness is less than the lower limit of the scheme roughness, the machining scheme can meet the requirements, but the economy is reduced, so its selectivity is zero. The membership function of the surface roughness is shown below

$$\left(R_a, x_j\right) = \begin{cases} 0 & R_a < a \\ 1 - \frac{(logb - logR_a)}{5(logb - loga)} & a \le R_a \le b \\ 0 & R_a > b \end{cases} \tag{24}$$

where, R_a is the surface roughness required by the machining feature, a is the minimum roughness that can be achieved by the machining scheme, and b is the maximum. Since the surface roughness standard series is approximately an equal ratio series, and its common ratio is $\sqrt{10}$, the logarithm of the surface roughness value is taken [25].

According to the membership function of the surface roughness $\mu\left(R_a, x_j\right)$, the fuzzy relationship between the surface roughness and the processing scheme set X can be obtained, which is expressed by the fuzzy matrix $R_{Ra\times X}$, as shown below

$$R_{Ra\times X} = \left[r_{1j}\right]_{1\times 6} \tag{25}$$

where, r_{1j} represents the possibility of selecting the machining scheme x_j when the machining surface roughness requirement is Ra.

5.4. Fuzzy Comprehensive Evaluation of Feature Processing Scheme

The fuzzy comprehensive evaluation of feature machining scheme is a decision-making process of the machining scheme, which comprehensively considers various influencing factors. According to the fuzzy relation matrix of each single-factor, the final fuzzy comprehensive evaluation matrix can be obtained, as shown in Equation (26).

$$R = \left[R_M, R_F, R_{IT}, R_{Ra}\right]^T \tag{26}$$

where, R_M is workpiece material fuzzy matrix, R_F is feature type fuzzy matrix, R_{IT} is machining accuracy fuzzy matrix, and R_{Ra} is surface roughness fuzzy matrix.

The calculation of fuzzy comprehensive evaluation index vector is shown in Formula (27).

$$B = W \cdot R \tag{27}$$

where, W is weight vector of influencing factors and R is fuzzy comprehensive evaluation matrix.

According to the calculation results, the candidate set element corresponding to the maximum evaluation index b_i is selected as the processing scheme. The pseudo-code of the feature processing scheme decision algorithm based on fuzzy comprehensive evaluation is shown in Appendix A Algorithm A6.

6. Process Route Optimization Based on MEDGraph

The process route optimization of the feature machining scheme is a problem of optimizing the Machining Element (ME) sequence. On the basis of analyzing the constraint relationship among the MEs, a model machining process route optimization method based on MEDGraph is proposed. First, MEDGraph is established based on the constraint relationship among the MEs, and then the topology sequence of all the MEs is listed by the improved topology sorting algorithm. Finally, the evaluation function is constructed with the machining cost or efficiency as the optimization objective to obtain the optimal process route.

6.1. Description of Process Route Optimization Problem

The optimization of the process route is to sort the processes in the machining scheme according to the process rules and optimization objectives and obtain the optimal or near-optimal processing route. A part to be machined can be regarded as a set of multiple machining features. It is assumed that a part has n machining features, which is represented as follows

$$Part = \{f_1, f_2, \cdots, f_n\} \tag{28}$$

where, f_i is the No.i machining feature in the part.

Each machining feature is usually formed through a plurality of processes, which correspond to the machining scheme of the machining feature. Dou et al. [40] defines the information entities related to feature processing as ME, which is represented as a six tuple, as shown below

$$me = (ID, f_i, S, M, T, TAD) \tag{29}$$

where, ID is the ME number; f_i is the characteristics processed by ME; S is the processing name, such as rough turning and fine turning; M is the machine tool used for processing; T is the required tools; and TAD is the tool approach direction.

The machining process of a part can be expressed as a set of MEs, as shown below

$$ME = \{me_1, me_2, \cdots, me_n\} \tag{30}$$

where, me_i is the No.i ME in the processing element set.

On the basis of the preceding definition, the optimization of the process route is transformed into the optimization of the processing elements sorting in ME. That is, the evaluation function is established with the processing cost or production efficiency as the optimization objective, and the processing element ranking result with the best target value is obtained under the condition that the relevant process rules are met.

Structure array me[] is constructed, which is used to store relevant information of processing elements. Its members include me_id, stage, TAD, feature_id, operation_name, t_id, and m_id, which are used to store ME ID, processing stage, TAD, machined feature, processing name, tool ID, and machine ID, respectively. Among them, the processing stages to which the processing element belongs are represented by 1, 2, 3, and 4, which are rough processing, semi-finishing, finish processing, and finishing processing, respectively.

6.2. The Construction of MEDGraph

In order to ensure the processing quality, the formulation of the process route needs to comply with the relevant rigid process rules and give priority to meeting the additional rules. Rigid process rules are mainly as follows: benchmark first, rough machining before finish machining, surface machining before hole, and process the main surface first and then the other surface. Additional rules are recommended rules during processing, mainly including: (1) Adjacency rule. The process execution sequence of certain machining features is preferably adjacent. (2) Clustering rules. Features on the same surface, features of the same type, or features that need to be processed with the same tool are best arranged for processing together.

Based on the constraint relationship between the processing elements, the process flow represented by the directed graph is called Machining Element Directed Graph (MEDGraph) [51], which is defined as

$$G = (V, E) \tag{31}$$

where, V is the node of the digraph, representing the MEs set, and its definition is shown as Formula (32); E is the set of directed edges, which indicates the priority relationship between MEs. Its definition is shown as Formula (33).

$$V = ME = \{me_0, me_1, me_2, \cdots, me_n\} \tag{32}$$

$$E = \{e(me_i, me_j) | 0 \le i, j \le n, i \ne j\} \tag{33}$$

The processing element directed graph is established according to the constraint relationship between processing elements. The establishment process is as follows: traverse all processing elements, randomly select two of them, create corresponding nodes, judge the constraint relationship between the two processing elements, and create a directed edge between nodes according to their priority order until all processing elements have created corresponding nodes. The process element digraph is represented by an adjacency matrix, and its definition is shown in Equation (34).

$$\mathrm{E}[n][n] = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & & e_{1n} \\ \vdots & & \ddots & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{nn} \end{bmatrix} \tag{34}$$

where, n is the number of nodes in MEDGraph, e_{ii} is machining element me_i, e_{ij} is the constraint relationship between me_i and me_j. The value rules of e_{ij} are as follows: when me_i takes precedence over me_j, $e_{ij} = 1$, $e_{ji} = -1$; if there is no priority constraint relationship between me_i and me_j, $e_{ij} = e_{ji} = 0$.

The construction process of the MEDGraph matrix is shown in Figure 20, and its main steps are as follows:

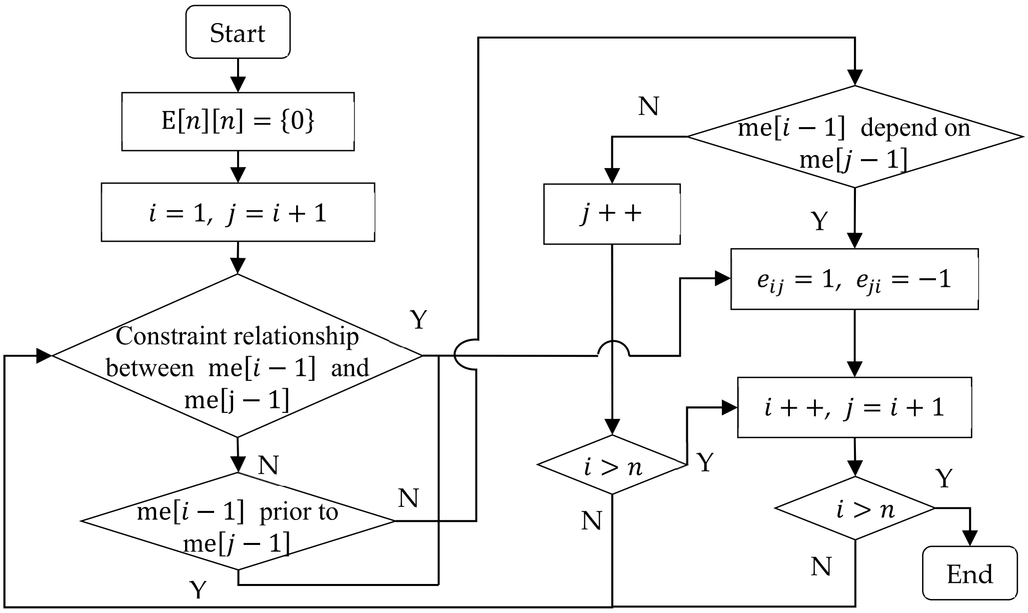

Start
E[n][n] = {0}
i = 1, j = i + 1
Constraint relationship between me[i − 1] and me[j − 1]
Y
N
me[i − 1] prior to me[j − 1]
N
Y
me[i − 1] depend on me[j − 1]
N
Y
j + +
e_{ij} = 1, e_{ji} = −1
i > n
Y
N
i + +, j = i + 1
i > n
Y
N
End

Figure 20. Construction of MEDGraph matrix.

(1) Assign an initial value of 0 to the element e_{ij} of $\mathrm{E}[n][n]$, $1 \le i \le n$, $1 \le j \le n$.

(2) Let $i = 1$, $j = i + 1$, and judge whether there is a benchmark relationship between me$[i-1]$.feature and me$[j-1]$.feature If me$[j-1]$.feature is based on me$[i-1]$.feature, let $e_{ij} = 1$, $e_{ji} = -1$; otherwise, go to (3).

(3) If me$[i-1]$.stage $<$ me$[j-1]$.stage, let $e_{ij} = 1$, $e_{ji} = -1$; otherwise, go to (4).

(4) Judge whether there is a dependency between me$[i-1]$.feature and me$[j-1]$.feature. If me$[j-1]$.feature depends on me$[i-1]$.feature, let $e_{ij} = 1$, $e_{ji} = -1$; otherwise, go to step 5.

(5) Let j + +, and return to (2) until j = n + 1 and go to (6).

(6) let i + +, j = i + 1, return to (2) until i = n + 1, and end the program.

6.3. Acquisition of ME Topological Sequence Based on Improved Topology Sorting Algorithm

The MEDGraph constructed by the preceding method is a Directed Acyclic Graph (DAG). Using a topological sorting algorithm, all vertices in the DAG can be arranged into a linear sequence so that any pair of nodes u and v in the graph, if $\langle u, v\rangle \in \mathrm{E(DAG)}$, u appears before v in the linear sequence. Such a linear sequence is called a topological sequence. The basic principles of the topology sorting algorithm are as follows: (1) select a node with a degree of 0 from the digraph and output it. (2) Delete this node and all directed edges from it from the directed graph. (3) Repeat the above steps until there are no more nodes with a degree of 0 in the digraph. The algorithm is shown in Figure 21.

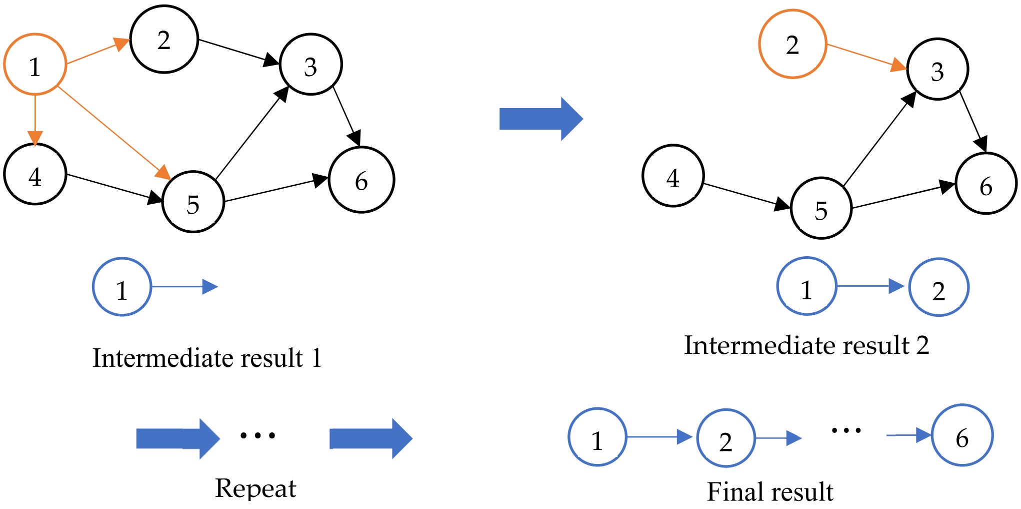


Figure 21. Topology sorting algorithm.

The results of the topological sorting of digraphs are not unique, and the conventional topological sorting algorithm can only obtain one result at random. By combining the topological sorting algorithm with the backtracking method, an improved topological sorting algorithm is proposed to obtain all the MEs topological sequences. The basic principle is as follows: after outputting a topological sequence, trace back to the previous node, and output nodes different from the previous topological sequence at this level. If there are no different nodes, continue to trace back to the previous level and finally realize the output of all topological sequences; its flow chart is shown in Figure 22, and the process is as follows:

(1) Read the MEDGraph.

(2) Find all the nodes in the current digraph with an indegree of 0, which constitute the node set $S(j)$, and set the set index $i = 0$. Go to (3) if the set is not empty; otherwise, go to 4.

(3) If $i < \mathrm{length}(S(j))$, take $S(j)[i]$ as the value of the current topological sequence element result$[j]$, and add 1 to i. Delete all the directional edges starting from result$[j]$, and mark the deleted directional edges so that they can be recovered during backtracking. Return to (2) and continue to obtain the value of the current topology sequence element result$[j+1]$.

If $i > \mathrm{length}(S(j))$, trace back to the last topological sequence element result$[j-1]$, and change its value to the next value in the corresponding set $S(j-1)$. Before using this value, recover all directed edges starting from $S(j-1)[n]$ and turn to (2). If n is equal to the number of elements in the set $S(j-1)$, the backtracking continues. Subtract 1 from j, and recover the deleted edge to obtain a new value. When j is equal to -1, it ends.

(4) If the number of elements in the current topology sequence is equal to the number of nodes, output it, go to (3), and continue to find other possible topology sequences.

Through the improved topological sorting algorithm, the topological sequence of all MEs is obtained.

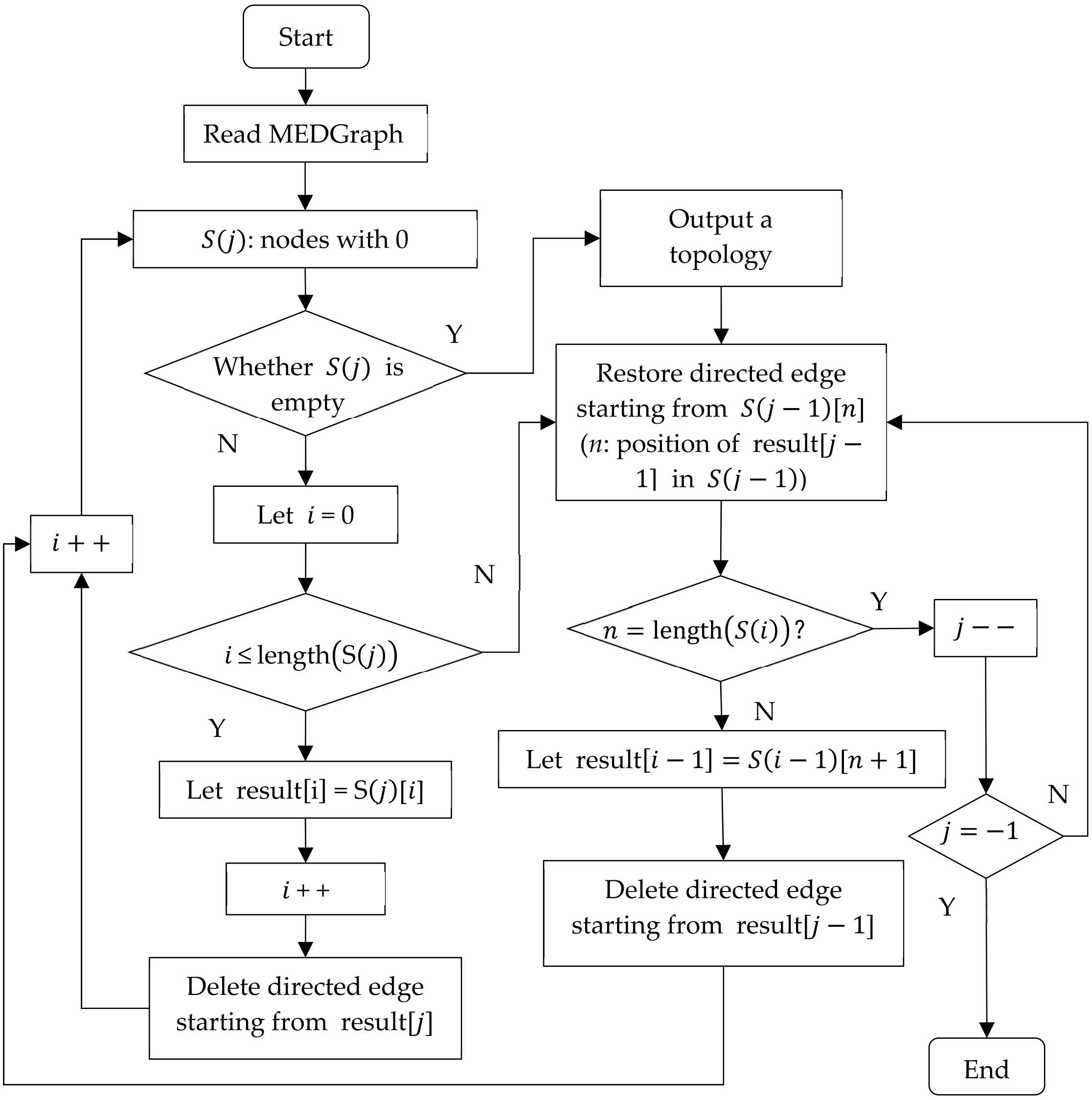

Start
Read MEDGraph
$S(j)$: nodes with 0
Whether $S(j)$ is empty
Y
N
Output a topology
Restore directed edge starting from $S(j-1)[n]$ (n: position of result[$j-1$] in $S(j-1)$)
Let $i = 0$
$i++$
N
$i \le \text{length}(S(j))$
Y
$n = \text{length}(S(i))$?
Y
$j--$
N
Let result[i] = S(j)[i]
Let result[$i-1$] = $S(i-1)[n+1]$
N
$j = -1$
$i++$
Delete directed edge starting from result[$j-1$]
Y
Delete directed edge starting from result[j]
End

Figure 22. Improved topology sorting algorithm.

6.4. Acquisition of Optimal Machining Element Topological Sequence

All the ME topological sequences are obtained by using the improved topological sorting algorithm mentioned above, and each of them can be converted into a feasible process route. An optimal route will be selected according to cost, efficiency, and other factors. Therefore, it is necessary to build an evaluation function and select the optimal ME topological sequence according to the value of the function. Considering that frequent replacement of manufacturing resources in the machining process will increase process error and reduce quality and efficiency, the evaluation function of the processing element sequence is constructed with the minimum number of replacements of manufacturing resources as the optimization objective, as follows

$$F = \gamma_1 F_1 + \gamma_2 F_2 + \gamma_3 F_3 \tag{35}$$

where, F_1, F_2, and F_3 are machine tool transformation function, fixture transformation function, and tool transformation function, respectively, and γ_1, γ_2, and γ_3 are the weight coefficients of the above functions.

Different ME may use different machine tools, but frequent replacement of the machine tools during the machining process will increase the auxiliary time. At the same time, because the workpiece needs to be relocated when replacing the machine tool, the machining accuracy will be affected. The MEs using the same machine tool shall be arranged in adjacent positions as far as possible. The machine transformation function is defined as follows:

$$F_1 = \sum_1^{n-1} \partial_1(\text{me}[i].\text{m\_id}, \text{me}[i+1].\text{m\_id}) \tag{36}$$

Similarly, different processing elements may use different fixtures and tools, and changing fixtures and tools will also increase auxiliary time and reduce accuracy. The processes using the same fixtures and tools shall be arranged in adjacent positions as far as possible. The fixture and tool transformation functions are shown in Formulas (37) and (38).

$$F_2 = \sum_1^{n-1} \partial_2(\partial_1(\text{me}[i].\text{m\_id}, \text{me}[i+1].\text{m\_id}), \partial_1(\text{me}[i].\text{TAD}, \text{me}[i+1].\text{TAD})) \tag{37}$$

$$F_3 = \sum_1^{n-1} \partial_2(\partial_1(\text{me}[i].\text{m\_id}, \text{me}[i+1].\text{m\_id}), \partial_1(\text{me}[i].\text{t\_id}, \text{me}[i+1].\text{t\_id})) \tag{38}$$

where, $\partial(x, y)$ is defined as follows:

$$\partial_1(x, y) = \begin{cases} 1 \; x \neq y \\ 0 \; x = y \end{cases} \tag{39}$$

$$\partial_2(x, y) = \begin{cases} 1 & \text{other} \\ 0 & x = y = 0 \end{cases} \tag{40}$$

The pseudo-code for obtaining the optimal machining element sequence is shown in Appendix A Algorithm A7.

7. Case Study

To validate the proposed machining feature recognition and process route optimization algorithm, a case study is performed in collaboration with a machinery enterprise. When developing product process flow, the enterprise first uses the machining feature recognition method proposed by us to traverse the machining features in the model, then makes the feature machining method decision, and finally optimizes the process route. We have established a prototype system, and its architecture is shown in Figure 23, which includes a human–computer interaction layer, functional layer, and database layer. The functions of each part are as follows:

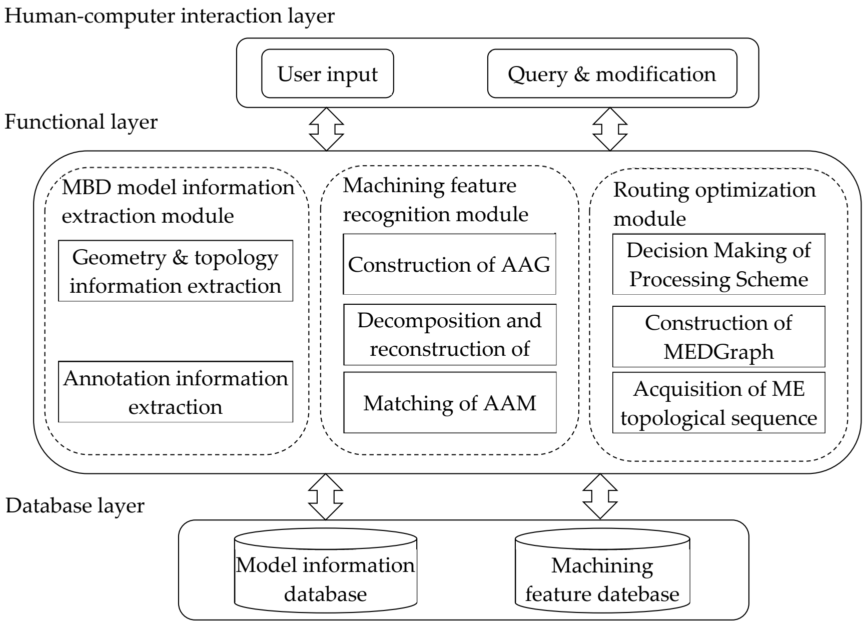

Human-computer interaction layer
User input
Query & modification
Functional layer
MBD model information extraction module
Geometry & topology information extraction
Annotation information extraction
Machining feature recognition module
Construction of AAG
Decomposition and reconstruction of
Matching of AAM
Routing optimization module
Decision Making of Processing Scheme
Construction of MEDGraph
Acquisition of ME topological sequence
Database layer
Model information database
Machining feature datebase

Figure 23. Prototype system architecture.

In order to verify the system functions, take the MBD model shown in Figure 24 as an example to verify the method proposed in this paper. First, extract the model information, as shown in Figures 25 and 26. The former figure shows the geometry and topology information of the model, and the latter shows the annotation information. A total of 71 faces and 219 edges of the model are extracted.

The extracted information is serialized in the Access database. With the tables of "surface_ data", "edge_ data", "dimension_ Data", and "gtol roughness", the database stores the model information of surface, edge, dimension, tolerance, and roughness information, respectively, as shown in Figure 27.

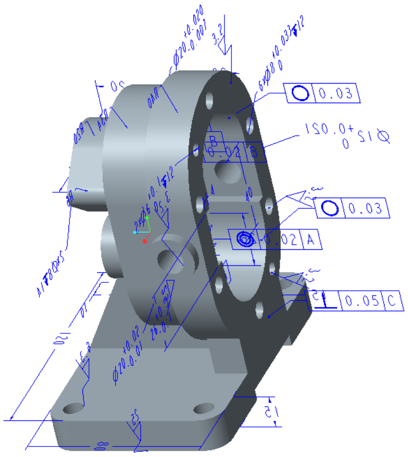
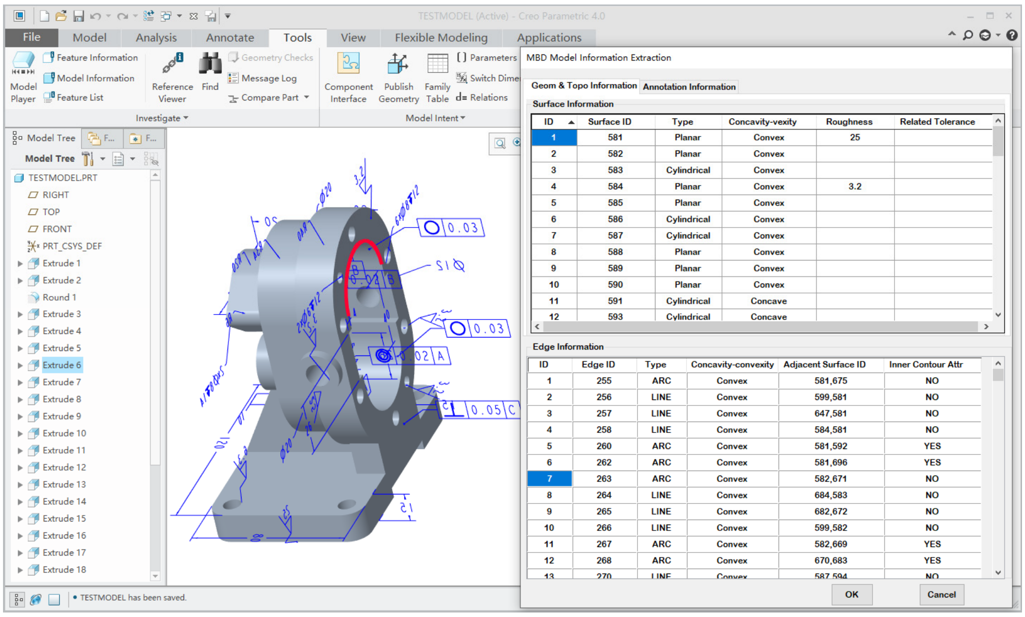


Figure 24. Example model.



• TESTMODEL has been saved.

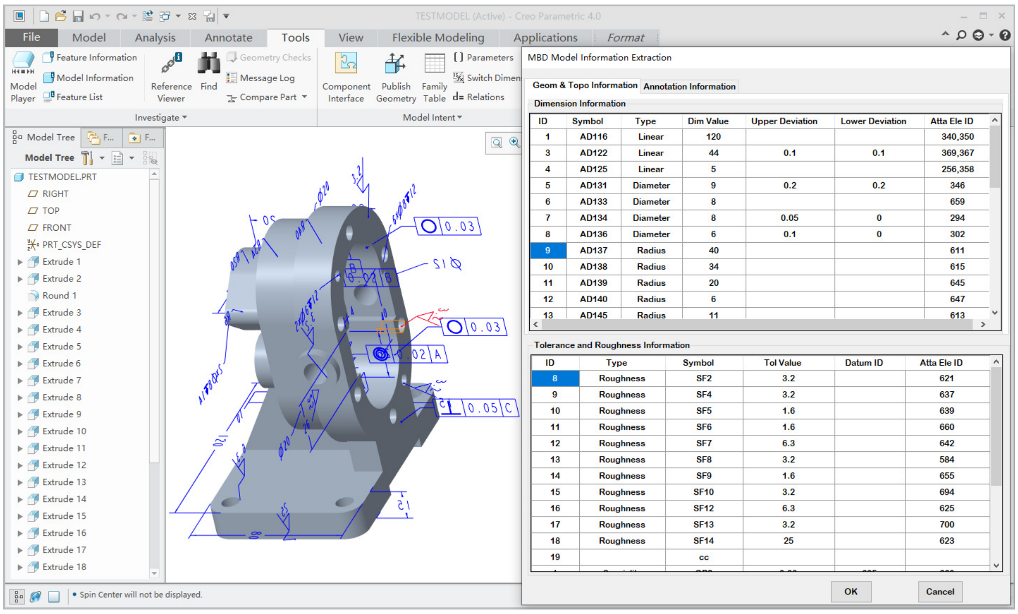
Surface Information

ID	Surface ID	Type	Concavity-vexity	Roughness	Related Tolerance
1	581	Planar	Convex	25	
2	582	Planar	Convex		
3	583	Cylindrical	Convex		
4	584	Planar	Convex	3.2	
5	585	Planar	Convex		
6	586	Cylindrical	Convex		
7	587	Cylindrical	Convex		
8	588	Planar	Convex		
9	589	Planar	Convex		
10	590	Planar	Convex		
11	591	Cylindrical	Concave		
12	593	Cylindrical	Concave		

Edge Information

ID	Edge ID	Type	Concavity-convexity	Adjacent Surface ID	Inner Contour Attr
1	255	ARC	Convex	581,675	NO
2	256	LINE	Convex	599,581	NO
3	257	LINE	Convex	647,581	NO
4	258	LINE	Convex	584,581	NO
5	260	ARC	Convex	581,592	YES
6	262	ARC	Convex	581,696	YES
7	263	ARC	Convex	582,671	NO
8	264	LINE	Convex	684,583	NO
9	265	LINE	Convex	682,672	NO
10	266	LINE	Convex	599,582	NO
11	267	ARC	Convex	582,669	YES
12	268	ARC	Convex	670,683	YES
13	270	LINE	Convex	587,594	NO

Figure 25. Extraction of model geometry and topology information.



MBD Model Information Extraction

Geom & Topo Information | Annotation Information

Dimension Information

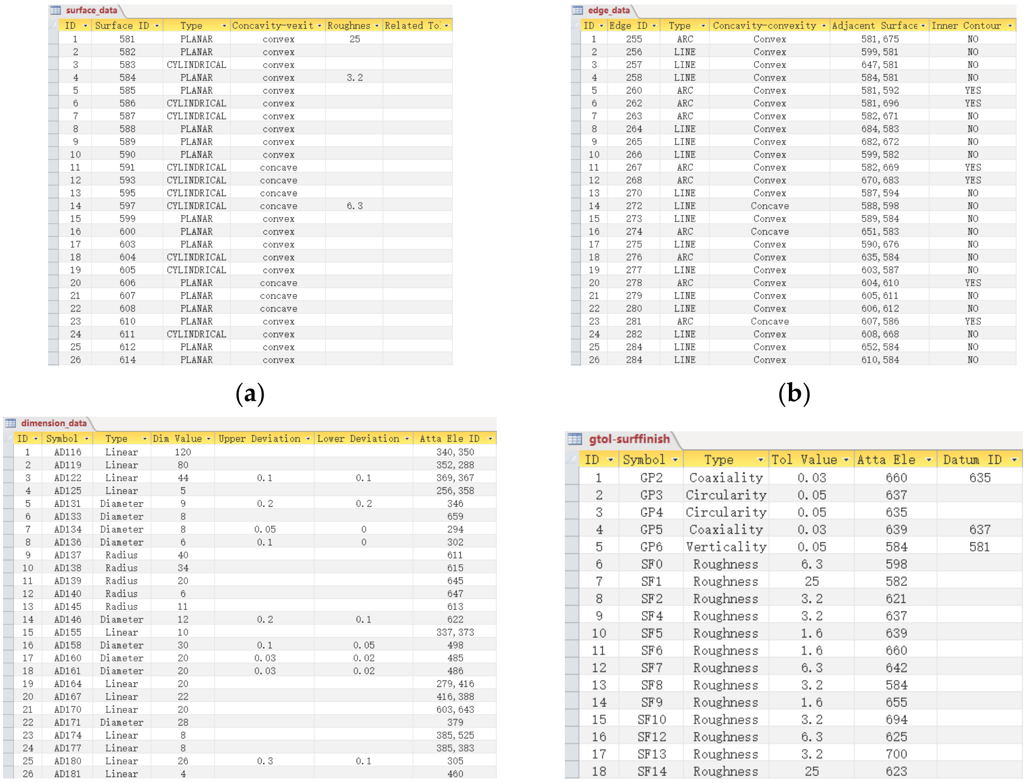
ID	Symbol	Type	Dim Value	Upper Deviation	Lower Deviation	Atta Ele ID
1	AD116	Linear	120			340,350
3	AD122	Linear	44	0.1	0.1	369,367
4	AD125	Linear	5			256,358
5	AD131	Diameter	9	0.2	0.2	346
6	AD133	Diameter	8			659
7	AD134	Diameter	8	0.05	0	294
8	AD136	Diameter	6	0.1	0	302
9	AD137	Radius	40			611
10	AD138	Radius	34			615
11	AD139	Radius	20			645
12	AD140	Radius	6			647
13	AD145	Radius	11			613

Tolerance and Roughness Information

ID	Type	Symbol	Tol Value	Datum ID	Atta Ele ID
8	Roughness	SF2	3.2		621
9	Roughness	SF4	3.2		637
10	Roughness	SF5	1.6		639
11	Roughness	SF6	1.6		660
12	Roughness	SF7	6.3		642
13	Roughness	SF8	3.2		584
14	Roughness	SF9	1.6		655
15	Roughness	SF10	3.2		694
16	Roughness	SF12	6.3		625
17	Roughness	SF13	3.2		700
18	Roughness	SF14	25		623
19		cc			

OK Cancel

Figure 26. Extraction of model annotation information.



surface_data

ID	Surface ID	Type	Concavity-vexit	Roughnes	Related To:
1	581	PLANAR	convex	25	
2	582	PLANAR	convex		
3	583	CYLINDRICAL	convex		
4	584	PLANAR	convex	3.2	
5	585	PLANAR	convex		
6	586	CYLINDRICAL	convex		
7	587	CYLINDRICAL	convex		
8	588	PLANAR	convex		
9	589	PLANAR	convex		
10	590	PLANAR	convex		
11	591	CYLINDRICAL	concave		
12	593	CYLINDRICAL	concave		
13	595	CYLINDRICAL	concave		
14	597	CYLINDRICAL	concave	6.3	
15	599	PLANAR	convex		
16	600	PLANAR	convex		
17	603	PLANAR	convex		
18	604	CYLINDRICAL	convex		
19	605	CYLINDRICAL	convex		
20	606	PLANAR	concave		
21	607	PLANAR	concave		
22	608	PLANAR	concave		
23	610	PLANAR	convex		
24	611	CYLINDRICAL	convex		
25	612	PLANAR	convex		
26	614	PLANAR	convex		

(**a**)

edge_data

ID	Edge ID	Type	Concavity-convexity	Adjacent Surface	Inner Contour
1	255	ARC	Convex	581, 675	NO
2	256	LINE	Convex	599, 581	NO
3	257	LINE	Convex	647, 581	NO
4	258	LINE	Convex	584, 581	NO
5	260	ARC	Convex	581, 592	YES
6	262	ARC	Convex	581, 696	YES
7	263	ARC	Convex	582, 671	NO
8	264	LINE	Convex	684, 583	NO
9	265	LINE	Convex	682, 672	NO
10	266	LINE	Convex	599, 582	NO
11	267	ARC	Convex	582, 669	YES
12	268	ARC	Convex	670, 683	YES
13	270	LINE	Convex	587, 594	NO
14	272	LINE	Concave	588, 598	NO
15	273	LINE	Convex	589, 584	NO
16	274	ARC	Concave	651, 583	NO
17	275	LINE	Convex	590, 676	NO
18	276	ARC	Convex	635, 584	NO
19	277	LINE	Convex	603, 587	NO
20	278	ARC	Convex	604, 610	YES
21	279	LINE	Convex	605, 611	NO
22	280	LINE	Convex	606, 612	NO
23	281	ARC	Concave	607, 586	YES
24	282	LINE	Convex	608, 668	NO
25	284	LINE	Convex	652, 584	NO
26	284	LINE	Convex	610, 584	NO

(**b**)

dimension_data

ID	Symbol	Type	Dim Value	Upper Deviation	Lower Deviation	Atta Ele ID
1	AD116	Linear	120			340, 350
2	AD119	Linear	80			352, 288
3	AD122	Linear	44	0.1	0.1	369, 367
4	AD125	Linear	5			256, 358
5	AD131	Diameter	9	0.2	0.2	346
6	AD133	Diameter	8			659
7	AD134	Diameter	8	0.05	0	294
8	AD136	Diameter	6	0.1	0	302
9	AD137	Radius	40			611
10	AD138	Radius	34			615
11	AD139	Radius	20			645
12	AD140	Radius	6			647
13	AD145	Radius	11			613
14	AD146	Diameter	12	0.2	0.1	622
15	AD155	Linear	10			337, 373
16	AD158	Diameter	30	0.1	0.05	498
17	AD160	Diameter	20	0.03	0.02	485
18	AD161	Diameter	20	0.03	0.02	486
19	AD164	Linear	20			279, 416
20	AD167	Linear	22			416, 388
21	AD170	Linear	20			603, 643
22	AD171	Diameter	28			379
23	AD174	Linear	8			385, 525
24	AD177	Linear	8			385, 383
25	AD180	Linear	26	0.3	0.1	305
26	AD181	Linear	4			460

(**c**)

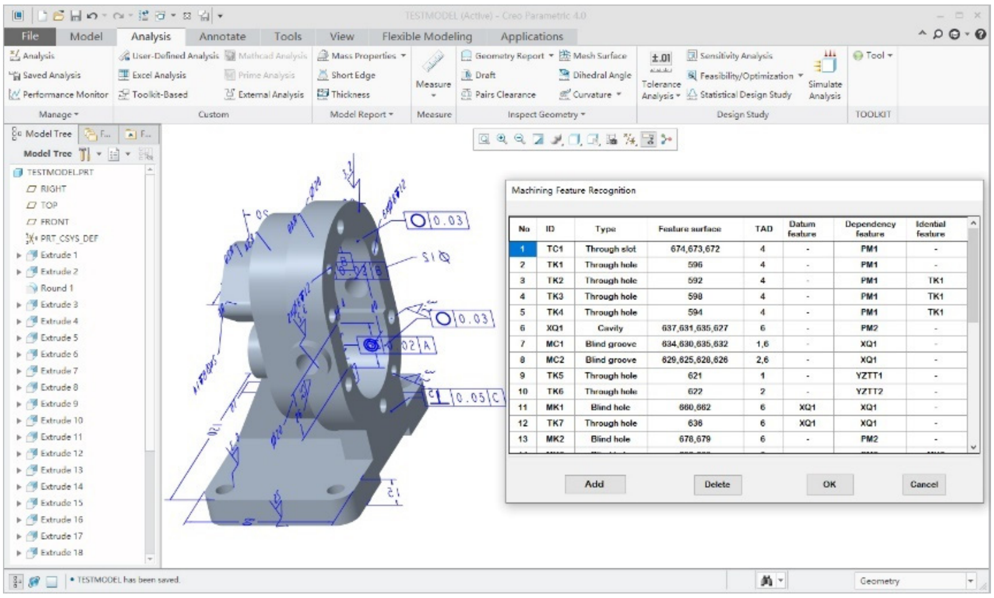
gtol-surffinish

ID	Symbol	Type	Tol Value	Atta Ele	Datum ID
1	GP2	Coaxiality	0.03	660	635
2	GP3	Circularity	0.05	637	
3	GP4	Circularity	0.05	635	
4	GP5	Coaxiality	0.03	639	637
5	GP6	Verticality	0.05	584	581
6	SF0	Roughness	6.3	598	
7	SF1	Roughness	25	582	
8	SF2	Roughness	3.2	621	
9	SF4	Roughness	3.2	637	
10	SF5	Roughness	1.6	639	
11	SF6	Roughness	1.6	660	
12	SF7	Roughness	6.3	642	
13	SF8	Roughness	3.2	584	
14	SF9	Roughness	1.6	655	
15	SF10	Roughness	3.2	694	
16	SF12	Roughness	6.3	625	
17	SF13	Roughness	3.2	700	
18	SF14	Roughness	25	623	

(**d**)

Figure 27. Storage of model information. (**a**) Surface information; (**b**) Edge information; (**c**) Dimension information; (**d**) Tolerance and roughness information.

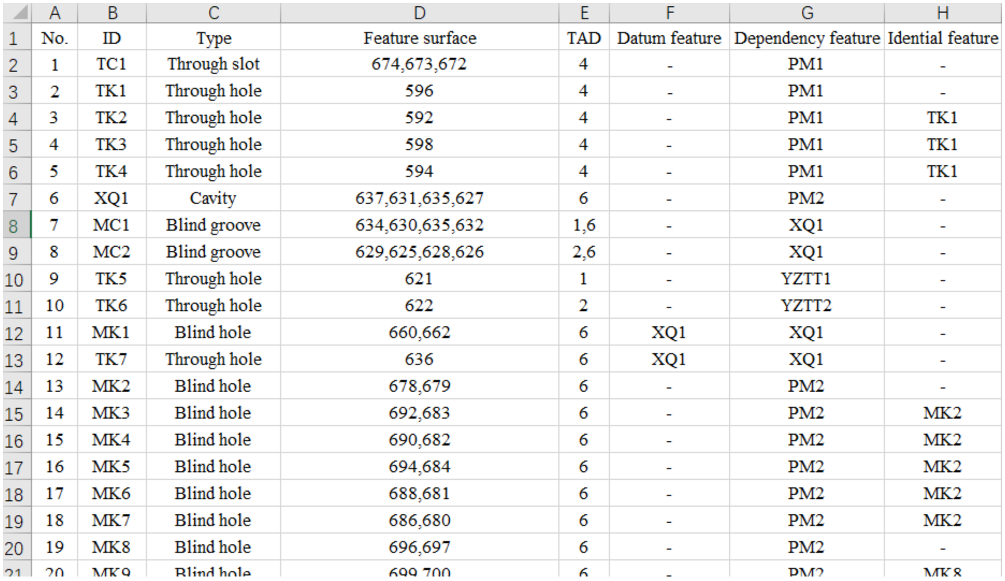
AAG decomposition method based on DBS and JBS is adopted to identify machining features, and the results are shown in Figure 28. A total of 28 machining features were identified, and the results were automatically imported into the Excel database to complete serialization, as shown in Figure 29.



Machining Feature Recognition

No	ID	Type	Feature surface	TAD	Datum feature	Dependency feature	Idential feature
1	TC1	Through slot	674,673,672	4	-	PM1	-
2	TK1	Through hole	596	4	-	PM1	-
3	TK2	Through hole	592	4	-	PM1	TK1
4	TK3	Through hole	598	4	-	PM1	TK1
5	TK4	Through hole	594	4	-	PM1	TK1
6	XQ1	Cavity	637,631,635,627	6	-	PM2	-
7	MC1	Blind groove	634,630,635,632	1,6	-	XQ1	-
8	MC2	Blind groove	629,625,628,626	2,6	-	XQ1	-
9	TK5	Through hole	621	1	-	YZTT1	-
10	TK6	Through hole	622	2	-	YZTT2	-
11	MK1	Blind hole	660,662	6	XQ1	XQ1	-
12	TK7	Through hole	636	6	XQ1	XQ1	-
13	MK2	Blind hole	678,679	6	-	PM2	-

Figure 28. Recognition of machining features.

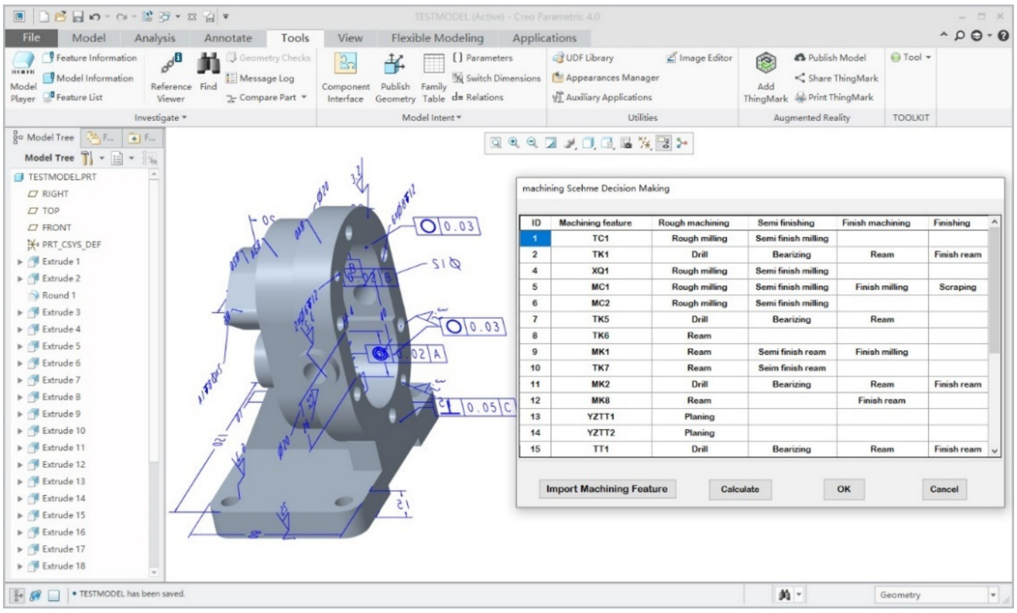


	A	B	C	D	E	F	G	H
1	No.	ID	Type	Feature surface	TAD	Datum feature	Dependency feature	Idential feature
2	1	TC1	Through slot	674,673,672	4	-	PM1	-
3	2	TK1	Through hole	596	4	-	PM1	-
4	3	TK2	Through hole	592	4	-	PM1	TK1
5	4	TK3	Through hole	598	4	-	PM1	TK1
6	5	TK4	Through hole	594	4	-	PM1	TK1
7	6	XQ1	Cavity	637,631,635,627	6	-	PM2	-
8	7	MC1	Blind groove	634,630,635,632	1,6	-	XQ1	-
9	8	MC2	Blind groove	629,625,628,626	2,6	-	XQ1	-
10	9	TK5	Through hole	621	1	-	YZTT1	-
11	10	TK6	Through hole	622	2	-	YZTT2	-
12	11	MK1	Blind hole	660,662	6	XQ1	XQ1	-
13	12	TK7	Through hole	636	6	XQ1	XQ1	-
14	13	MK2	Blind hole	678,679	6	-	PM2	-
15	14	MK3	Blind hole	692,683	6	-	PM2	MK2
16	15	MK4	Blind hole	690,682	6	-	PM2	MK2
17	16	MK5	Blind hole	694,684	6	-	PM2	MK2
18	17	MK6	Blind hole	688,681	6	-	PM2	MK2
19	18	MK7	Blind hole	686,680	6	-	PM2	MK2
20	19	MK8	Blind hole	696,697	6	-	PM2	-
21	20	MK9	Blind hole	699,700	6	-	PM2	MK8

Figure 29. Serialization of recognition result.

Set the material attribute of the model as cast iron, and use the feature processing scheme decision based on fuzzy comprehensive evaluation to make the feature processing scheme decision. The result is shown in Figure 30. Serialize the decision results.

According to the machining feature recognition and processing scheme decision results, the ME sequence is generated and stored in the database, as shown in Figure 31. The manufacturing resource information involved is shown in Table 11.



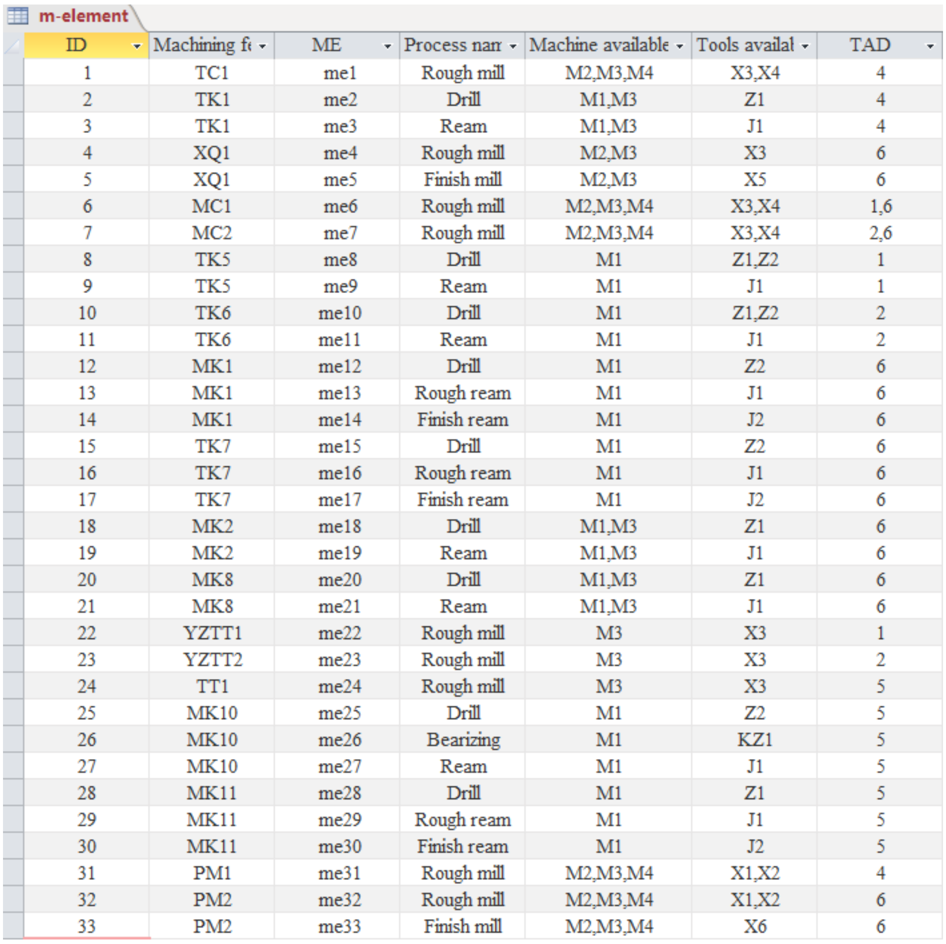
machining Scehme Decision Making

ID	Machining feature	Rough machining	Semi finishing	Finish machining	Finishing
1	TC1	Rough milling	Semi finish milling		
2	TK1	Drill	Bearizing	Ream	Finish ream
4	XQ1	Rough milling	Semi finish milling		
5	MC1	Rough milling	Semi finish milling	Finish milling	Scraping
6	MC2	Rough milling	Semi finish milling		
7	TK5	Drill	Bearizing	Ream	
8	TK6	Ream			
9	MK1	Ream	Semi finish ream	Finish milling	
10	TK7	Ream	Seim finish ream		
11	MK2	Drill	Bearizing	Ream	Finish ream
12	MK8	Ream		Finish ream	
13	YZTT1	Planing			
14	YZTT2	Planing			
15	TT1	Drill	Bearizing	Ream	Finish ream

Import Machining Feature Calculate OK Cancel

• TESTMODEL has been saved.

Figure 30. Decision results of machining scheme.



m-element

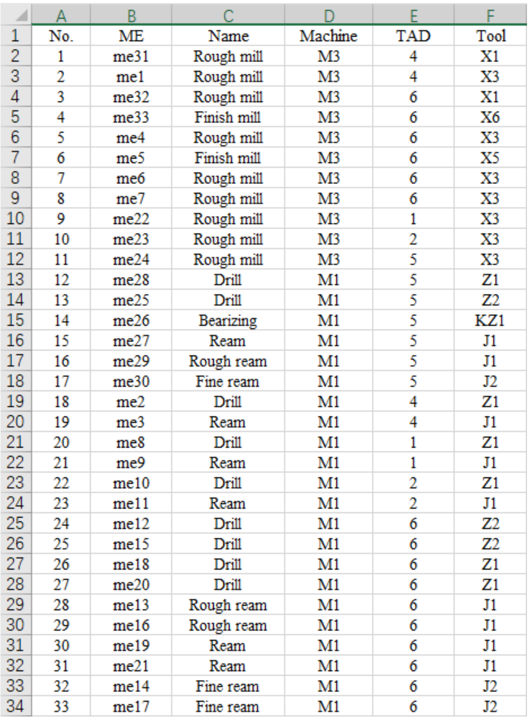
ID	Machining fe	ME	Process nan	Machine available	Tools availal	TAD
1	TC1	me1	Rough mill	M2,M3,M4	X3,X4	4
2	TK1	me2	Drill	M1,M3	Z1	4
3	TK1	me3	Ream	M1,M3	J1	4
4	XQ1	me4	Rough mill	M2,M3	X3	6
5	XQ1	me5	Finish mill	M2,M3	X5	6
6	MC1	me6	Rough mill	M2,M3,M4	X3,X4	1,6
7	MC2	me7	Rough mill	M2,M3,M4	X3,X4	2,6
8	TK5	me8	Drill	M1	Z1,Z2	1
9	TK5	me9	Ream	M1	J1	1
10	TK6	me10	Drill	M1	Z1,Z2	2
11	TK6	me11	Ream	M1	J1	2
12	MK1	me12	Drill	M1	Z2	6
13	MK1	me13	Rough ream	M1	J1	6
14	MK1	me14	Finish ream	M1	J2	6
15	TK7	me15	Drill	M1	Z2	6
16	TK7	me16	Rough ream	M1	J1	6
17	TK7	me17	Finish ream	M1	J2	6
18	MK2	me18	Drill	M1,M3	Z1	6
19	MK2	me19	Ream	M1,M3	J1	6
20	MK8	me20	Drill	M1,M3	Z1	6
21	MK8	me21	Ream	M1,M3	J1	6
22	YZTT1	me22	Rough mill	M3	X3	1
23	YZTT2	me23	Rough mill	M3	X3	2
24	TT1	me24	Rough mill	M3	X3	5
25	MK10	me25	Drill	M1	Z2	5
26	MK10	me26	Bearizing	M1	KZ1	5
27	MK10	me27	Ream	M1	J1	5
28	MK11	me28	Drill	M1	Z1	5
29	MK11	me29	Rough ream	M1	J1	5
30	MK11	me30	Finish ream	M1	J2	5
31	PM1	me31	Rough mill	M2,M3,M4	X1,X2	4
32	PM2	me32	Rough mill	M2,M3,M4	X1,X2	6
33	PM2	me33	Finish mill	M2,M3,M4	X6	6

Figure 31. MEs sequence.

Table 11. Manufacturing resource information.

No.	Type	Name	Resource	No.	Type	Name	Resource
1	machine	M1	drilling machine	1	tool	Z1, Z2, Z3	Twist drill
2		M2	Vertical milling machine	2		J1, J2	Reamer
3		M3	CNC milling machine	3		KZ1	Counterbore drill
4		M4	Horizontal milling machine	4		X1, X2, X3, X4, X5, X6	Milling cutter

Fifty-three topological sequences of ME are obtained by optimizing the ME sorting, among which the evaluation function value of the optimal sequence: F = 1 × 0.530 + 9 × 0.297 + 21 × 0.163 = 6.626; this sequence is the optimal process route for this part machining, as shown in Figure 32.



No.	ME	Name	Machine	TAD	Tool
1	me31	Rough mill	M3	4	X1
2	me1	Rough mill	M3	4	X3
3	me32	Rough mill	M3	6	X1
4	me33	Finish mill	M3	6	X6
5	me4	Rough mill	M3	6	X3
6	me5	Finish mill	M3	6	X5
7	me6	Rough mill	M3	6	X3
8	me7	Rough mill	M3	6	X3
9	me22	Rough mill	M3	1	X3
10	me23	Rough mill	M3	2	X3
11	me24	Rough mill	M3	5	X3
12	me28	Drill	M1	5	Z1
13	me25	Drill	M1	5	Z2
14	me26	Bearizing	M1	5	KZ1
15	me27	Ream	M1	5	J1
16	me29	Rough ream	M1	5	J1
17	me30	Fine ream	M1	5	J2
18	me2	Drill	M1	4	Z1
19	me3	Ream	M1	4	J1
20	me8	Drill	M1	1	Z1
21	me9	Ream	M1	1	J1
22	me10	Drill	M1	2	Z1
23	me11	Ream	M1	2	J1
24	me12	Drill	M1	6	Z2
25	me15	Drill	M1	6	Z2
26	me18	Drill	M1	6	Z1
27	me20	Drill	M1	6	Z1
28	me13	Rough ream	M1	6	J1
29	me16	Rough ream	M1	6	J1
30	me19	Ream	M1	6	J1
31	me21	Ream	M1	6	J1
32	me14	Fine ream	M1	6	J2
33	me17	Fine ream	M1	6	J2

Figure 32. Optimal ME sequence.

8. Conclusions

In summary, aiming at the problem of machining feature recognition and process route optimization in the MBD model, we propose a feature matching algorithm based on model AAM after studying the model AAG decomposition based on DBS and JBS, which first constructs the decomposed feature subgraph as AAM, and then realizes machining feature recognition by matching with predefined feature AAM. A feature processing scheme decision-making method based on fuzzy comprehensive evaluation is proposed. By analyzing the influencing factors of processing scheme selection, the mapping relationship between the influencing factors and the processing scheme is established. Based on the single-factor evaluation results, a multi-factor comprehensive evaluation model is established. By calculating the comprehensive evaluation index, the optimization of the feature processing scheme is realized. Based on the optimized scheme, a model processing route optimization method based on MEDGraph is proposed. Based on the constraint relationship between MEs, the MEDGraph is established. The improved topological sorting algorithm is applied to list the topological sequence of all MEs. Finally, the evaluation function is constructed to obtain the optimal process route. Our research realized the recognition of machining features in parts and the determination of the optimal process route without complex similarity transformation of AAM. The application of our research can provide a premise for enterprise intelligent manufacturing on the basis of automatic recognition of model machining features.

Our future work will focus on the recognition of machining features, including free-form surfaces and complex intersection features, as well as the optimization of process routes considering other factors, such as machining parameters and process tolerance allocation.

Author Contributions: S.D. made contributions to the conceptualization and original draft writing. Z.G. contributed to the validation and resource. B.W. contributed to the original draft writing and software prototype validation. H.W. contributed to the investigation and editing. F.M. contributed to the draft review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 52005302 and No.52105463).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Algorithm A1. Geometric & topology information extraction algorithm

```
Input: MBD model
Output: Store geometric & topology information in computer memory
/*Definition of structure array*/
struct Surface_data
{
    ElemType surf_id
    ElemType surf_type
    ElemType surf_at
    vector< ElemType > adj_surf_id
    vector< ElemType > edge_id
    vector< ElemType > cont_data
    ElemType surffinish_value
    vector< ElemType > gtol_data
}
struct Contour_data
{
    vector< ElemType > edge_id
    ElemType contour_type
}
struct Edge_data
{
    ElemType edge_id
    ElemType edge_type
    ElemType edge_at
    vector<ElemType > adja_suf
    ElemType edge_in
}
```

Algorithm A1. *Cont.*

```
/*Define structure array*/
Surface_data surf[num1], Contour_data contour[num2], Edge_data edge[num3]
k← 0, h← 0
Call ProMdlCurrentGet( ) to get the handle of the current model
Call ProSolidSurfaceVisit( ) to traverse surface and obtain and output surface[]
Call ProArraySizeGet( ) to get the length of surf[ ], which is m
for (i = 0; i < m; i ++) do
    Call ProSurfaceIdGet( ) to get the ID of No.i+1 surface and write it to surf[i].surf_id
    Call ProSurfaceTypeGet( ) to get the type of No.1+1 surface and write to surf[i]. surf_type
    Call ProSurfaceAtGe ( ) to get the concavity-convexity of No.1+1 surface
    Write to surface[i].surf_at
    Call ProSurfaceContourVisit( ) to traverse the contour and write them to Contour[ ]
    Call ProArraySizeGet( ) to get the length of Contour[], which is n
    for (j = 0; j < n; j ++) do
        Call ProContourTraversalGet( ) to get the type of contour and write to contour[k].contour_type
        Call ProContourEdgeVisit( ) to traverse the edges of contour and write to Edge[]
        Call ProArraySizeGet( ) to get the length of edge[], which is e
        for (t = 0; t < e; t ++) do
            Call ProEdgeIdGet( ) to get the ID of edge
            Write it to edge[h].edge_id, contour [k].edge_id
            if (edge[h].edge_id ∉ surface[i].edge_id ) then
                Write to surface[i].edge_id
            end if
            Call ProEdgeTypeGet( ) to get type of edge and write to edge[h].edge_type
            Call ProEdgeAtGet( ) to get concavity-convexity and write to edge[h].edge_at
            Call ProEdgeNeighborsGet( ) to get the adjacent surface ID
            Write to edge[h].adja_suf
            if (edge[h].adja_suf ∉ surface[i].edge_id ) then
                Write to surface[i]. adj_surf_id
            end if
        h++
        end for
        Write contour[k] to surface[i].cont_data
        k++
        end for
end for
```

Algorithm A2. AAG construction

```
Input: MBD model information structure array surface[ ], edge[ ] and contour[ ]
Output: line[num1], node[num2]
/*Define structure array node[num2] and line[num1], num1 and num2 is the number of model edge and surface*/
struct Node
{
    ElemType surface_id
    vector< ElemType > adj_line
    ElemType node_type
    ElemType node_at
    ElemType d_face
```

Algorithm A2. *Cont.*

```
    ElemType f_face
    ElemType c_face
} node[num2]
struct Line
{
    ElemType edge_id
    ElemType node1
    ElemType node2
    ElemType line_type
    ElemType line_at
    ElemType in_contour
} line[num1]
/*Assign the id of line and surface */
for (i = 0; i < num1; i ++) do
    line[i]. edge_id ← edge[i].edge_id
end for
for (i = 0; i < num2; i ++) do
    node[j]. surface_id ← surface[j].surface_id
end for
/* Determine the connection relationship between nodes and edges*/
for (i = 0; i < num2; i ++) do
    for (j = 0; j < num1; j ++) do
        for (k = 0; k < surface[i].edge_id.size(); k ++) do
            if (line[j]. edge_id == surface[i].edge_id[k]) then
                Add line[j] to node[i]. adj_line
                if (line[j].node1==NULL) then
                    line[j].node1 ← node[i]
                else
                    line[j].node2 ← node[i]
                end if
            end if
        end for
    end for
end for
/*Add node attribute*/
for (i = 0; i < num2; i ++) do
    /*Assign the type*/
    if (surface[i].type == planar) then node[i].node_type ← 1 end if
    if (surface[i].type == cylinder) then node[i].node_type ← 2 end if
    if (surface[i].type == conical) then node[i].node_type ← 3 end if
    if (surface[i].type == other) then node[i].node_type ← 4 end if
    /*Assign the concavity-convexity of surface*/
    if (surface[i].concavity-convexity ==concave) then
        node[i].node_at ← −1
    else
        node[i].node_at ← 1
    end if
    /*Assign the DBS attribute*/
    if (surface[i].contour_information == inner) then
        node[i].d_face ← true
    else
        node[i].d_face ← false
    end if
    /*Assign the FBS attribute*/
    Calculation the extrusion of $f_i$, which is $V_b$
```

Algorithm A2. *Cont.*

```
    if (surface[i].type == planar && surface[i].concavity-convexity == convex && $V_b \cap\ part = \varnothing$ then
        node[i].f_face ← true
    else
        node[i].f_face ← false
    end if
end for
/*Traverse the JBS*/
for (i = 0; i < num2; i ++) do
    for (j = 0; j < num2; j ++) do
        if (surface[i] and surface[j] == JBS) then
            node[i].c_face ← node[j]. c_face ← k
            k++
        end if
    end for
end for
/*Add edge attribute*/
for (i = 0; i < num1; i ++) do
    /*Assign the type*/
    if (edge[i].type == straight) then line[i].line_type ← 1 end if
    if (edge[i].type == arc) then line[i].line_type ← 2 end if
    if (edge[i].type == other) then line[i].line_type ← 3 end if
    if (edge[i].type == straight) then line[i].line_type ← 1 end if
    /*Assign the concavity-convexity*/
    if (edge[i].concavity-convexity == concavity) then line[i].line_at ← $-1$ end if
    if (edge[i].concavity-convexity == convexity) then line[i].line_at ← 1 end if
    if (edge[i].concavity-convexity == smooth) then line[i].line_at ← 0 end if
    /*Assign the inner contour attribute*/
    if (edge[i].inner_contour == true) then
        line[i]. in_contour ← true
    else
        line[i]. in_contour ← false
    end if
end for
```

Algorithm A3. AAG decomposition and reconstruction based on DBS and JBS

```
Input: line[num1] and node[num2]
Output: line1[n][ ] and node1[n][ ], which represent the edge and node after reconstruction
/*Decomposition based on DBS*/
Nsign[i] ← 0
for (i = 0; i < num2; i ++) do
    if (node[i].d_face == ture) then
        for (j = 0; j < num1; j ++) do
            if (line[j].node1 == node[i] | | line[j].node2 = node[i]) then
                if (line[j].in_contour == true && line[j].line_at == $-1$) then
                    Nsign[i] ← $-1$
                    line[j] ← NULL
                end if
                if (line[j].in_contour == false && line[j].line_at == 1) then
                    Nsign [i] ← 1
                    line[j] ← NULL
                end if
            end if
        end for
    end if
end for
```

Algorithm A3. *Cont.*

```
for ($i = 0; i <$ num2; $i$ ++ ) do
    if (node[$i$] $==-1$) then
        Extract the information of node[ ] and line[ ] adjacent to node[i]
        if (line[ ].$line\_at = -1$) then
            line[ ] ← NULL
            Assign the value of node[ ] to corresponding node1[ ][ ]
            Obtain the value of line1[ ][ ] according to node1[ ][ ]
        else
            Store the corresponding node1[ ][ ] and line1[ ][ ]
        end if
    end if
    if (node[$i$] ==1) then
        Extract the information of node[ ] and line[ ] adjacent to node[i], and store them to node1[ ][ ] and line1[ ][ ]
    end if
    Delete line[ ] and node[ ]
end for
/*Delete non-machining surface*/
for ($i = 0; i <$ node[ ].size; $i$ ++) do
    if (node[$i$].$node\_at == 1$) then
        node[$i$] ← NULL
        Delete line[ ] adjacent to node[i].adj_line
        Update line[ ] and node[ ]
    end if
end for
/*Extract the MCCS on FBS*/
for ($j = 0; j <$ node[ ].size; $j$ ++) do
    if (node[$j$].$f\_face$ == true) then
        for ($k = 0; k <$ node[$j$].adj_line.size; $k$ ++) do
            if (node[$j$].adj_line[$k$].$line\_at == -1$) then
                Extract the node adjacent to node[$j$].adj_line[$k$]
                if (extracted node has concave edge) then
                    Continue to extract nodes adjacent to the concave edges
                end if
            end if
        end for
    end if
    Store node[ ] to node1[ ][ ] and then get the corresponding line1[ ][ ]
end for
/*Search the node with the same JBS attribute*/
c_sign[n][ ] ← 0
for ($i = 0; i <$ n; $i$ ++) do
    for ($j = 0; j <$ node1[$i$][ ].size; $j$ ++) do
        if (node1[$i$][$j$].$c\_face \neq \varnothing$ then
            c_sign[$i$][$j$] ←node1[$i$][$j$].$c\_face$
        end if
    end for
end for
for ($k = 0; k <$ n; $k$ ++) do
    for ($s = 0; s <$ n; $s$ ++) do
        if (c_sign[$k$][ ] and c_sign[$s$][ ] have the same element) then
            Merge line1[$s$][ ] to line1[$k$][ ]
            Merge node1[$s$][ ] to node1[$k$][ ]
        end if
    end for
end for
```

Algorithm A4. AAM establishment algorithm

```
Input: line2[m][ ] and node2[m][ ]
Output: The AAM of machining feature to be identified
$sg \leftarrow 0$, M[$n$][$n$] $\leftarrow$ 0
for ($t$ = 0; $t$ < m; $t$ ++) do
    for ($i$ = 0; $i$ < node[$t$][ ].size; $i$ ++) do
      a $\leftarrow$ 2, $b \leftarrow c \leftarrow 0$
      c $\leftarrow$ node2[$t$][$i$].node_at
      b $\leftarrow$ node2[$t$][$i$].node_type
      if (node2[$t$][$i$].node_ f_face == true) then
        $a \leftarrow 1$
       end if
       M[$sg$][$sg$] $\leftarrow$ $a$*100+$b$*10+$c$
      $sg$++
  end for
end for
for ($w$ = 0; $w$ < m; $w$ ++) do
  for ($j$ = 0; $j$ < line2[$w$][ ].size; $j$ ++) do
       d $\leftarrow$ 0, e $\leftarrow$ 0
       d $\leftarrow$ line2[$w$][$j$].$line\_type$
       e $\leftarrow$ line2[$w$][$j$].$line\_at$
       Search node2[ ][ ] for line2[$w$][$j$].$node1$
       Search node2[ ][ ] for line2[$w$][$j$].$node2$
       Determine the position of line2[$w$][$j$] in AAM according to node2[ ][ ]
       M[$k$][$h$] $\leftarrow$ M[$h$][$k$] $\leftarrow$ d*10+e
     end for
end for
```

Algorithm A5. Matching between feature subgraph AAM and predefined feature AAM

```
Input: feature subgraph AAM to be identified represented by matrix A[n][n], and predefined
feature AAM represented by matrix B[m][m]
Output: matching result
/* Define the function SetJudgment(matrix1, matrix2) to determine whether matrix1 and
matrix2 is identical */
SetJudgment(matrix1, matrix2)
{
    $k \leftarrow 0$
    for ($i$ = 0; $i$ < matrix1.size; $i$ ++) do
      for ($j$ = 0; $j$ < matrix2.size; $j$ ++)j $\leftarrow$ 0 to matrix2.size do
         if (matrix1[$i$] == matrix2[$j$]) then
           $k$++
           matrix2 $\leftarrow$ matrix2.erase(matrix2.begin( )+$j$)
            break
          end if
      end for
    end for
      if ($k$ == $n$) then
          result $\leftarrow$ true
      else
         result $\leftarrow$ false
      end if
      return result
}
for ($i$ = 0; $i$ < f; $i$ ++) do//$f$ is the number of feature subgraphs
      if ($n \neq m$) then
```

Algorithm A5. *Cont.*

```
        Get next AAM
    else
        vector<int> $S_1$, vector<int> $S_2$
        for ($i$ = 0; $i$ < n; $i$ ++) do
            $S_1 \leftarrow A[i][i]$
            $S_2 \leftarrow B[i][i]$
        end for
        result1 = SetJudgment($S_1$, $S_2$)
        if (!result1) then
            Get next AAM
        else
            vector<int> $S_{Ai}$, vector<int> $S_{Bi}$
            for ($i$ = 0; $i$ < n; $i$ ++) do
                $S_{Ai} \leftarrow A[i][\ ]$
                $S_{Bi} \leftarrow B[i][\ ]$
            end for
            $index \leftarrow 0$
            for ($w$ = 0; $w$ < n; $w$ ++) do
                for ($u$ = 0; $u$ < n; $u$ ++) do
                    result2=SetJudgment($S_{Aw}$, $S_{Bu}$)
                    if (!result2) then
                        Get next AAM
                    else
                        $S_{Bu} \leftarrow 0$
                        $index$ ++
                        break
                    end if
                end for
            end for
            if ($index \neq$ n) then
                Get next AAM
            else
                success
        end if
    end if
end for
```

Algorithm A6. Decision algorithm of feature machining scheme

```
Input: machining feature information
Output: Optimal processing scheme
Call ProMaterialDataGet( ) to get the material properties, which is $Mat$
for ($i$ = 0; $i$ < $n$; $i$ ++) do//$n$ is the number of processing features
    if (feature[$i$] $\neq$ hole feature) then//Processing scheme decision
        /* Calculate fuzzy matrix of each influencing factor for surface features*/
        for ($j$ = 0; $j$ < 6; $j$ ++) do
            if (x[$j$] is suitable for $Mat$) then RM[$j$] $\leftarrow$ 1 end if
            if (x[$j$] is suitable for feature[$i$].feature_type) then RF[$j$] $\leftarrow$ 1 end if
            GetSchemeIT(x[$j$])//Obtain the precision range ITn~ITm of scheme x[$j$]
                if (feature[$i$].IT $<$ ITn $-$ 2 || feature[$i$].IT $>$ ITm) then RIt[$j$] $\leftarrow$ 0 end if
                if (ITn $\leq$ feature[$i$].IT $\leq$ ITm) then RIt[$j$] $\leftarrow$ 1 end if
                    if (ITn $-$ 2 $\leq$ feature[$i$].IT $<$ ITn) then RIt[$j$] $\leftarrow$ 1-((ITn-feature[$i$].IT)/(5*(ITm-ITn))) end if
                    GetSchemeRa(x[$j$])//Obtain the roughness range a~b of scheme x[$j$]
                if feature[$i$].Ra $<$ a || feature[$i$].Ra $>$ b then RRa[$j$] $\leftarrow$ 0 end if
```

Algorithm A6. *Cont.*

```
            if a ≤ feature[i].Ra ≤ b then RRa[j] ← 1-(log(b)-log(feature[i].Ra))/5(log(b)-log(a)) end if
      end for
    /*Calculating fuzzy comprehensive evaluation matrix R1[ ][ ]*/
      for (k1 = 0; k1 < 6; k1 ++) do
        R1[0][k1] ← RM[k1]
      end for
      for (k2 = 0; k2 < 6; k2 ++) do
        R1[1][k2] ← RF[k2]
      end for
      for (k3 = 0; k3 < 6; k3 ++) do
        R1[2][k3] ← RIt[k3]
      end for
      for (k4 = 0; k4 < 6; k4 ++) do
        R1[3][k4] ← RRa[k4]
      end for
    /*Calculating fuzzy comprehensive evaluation index vector B[ ]*/
      for (num = 0; num < 6; num ++) do
        for (i = 0; i < 6; i ++) do
          for (j = 0; j < 4; j ++) do
            B[num] ←B[num]+W[j]*R1[i][j]
          end for
        end for
      end for
      Take x[ ] corresponding to element with largest value in B[ ] as the scheme [i]
    end if
    if (feature[i] == hole feature) then
      for (w = 0; w < 13; w ++) do
        if (y[w] is suitable for Mat) then RM[w] ← 1 end if
        GetFeatureD( )//Get the diameter D of hole feature
        if (y[w] is suitable for D) then RD[w] ← 1 end if
        GetSchemeIT(y[w]) //Get the precision range ITn ∼ ITm of scheme y[w]
        if (feature[i].IT < ITn − 2 || feature[i].IT > ITm) then RIt[j] ← 0 end if
        if (ITn ≤ feature[i].IT ≤ ITm) then RIt[j] ← 1 end if
          if (ITn − 2 ≤ feature[i].IT < ITn) then RIt[j] ← 1 − ((ITn − feature[i].IT)/(5 ∗ (ITm − ITn))) end if
          GetSchemeRa(y[w]) //Get the roughness range a ∼ b of scheme y[w]
      end if
      if (feature[i].Ra <  a || feature[i].Ra > b) then RRa[w] ← 0 end if
      if (feature[i].Ra <  a || feature[i].Ra > b) then
        RRa[w] ← 1-(log(b)-log(feature[i].Ra))/5(log(b)-log(a))
      end if
      end for
    /* Calculate fuzzy comprehensive evaluation matrix R2[ ][ ] of hole feature*/
      for (s1 = 0; s1 < 13; s1 ++) do
        R2[0][s1] ← RM[s1]
      end for
      for (s2 = 0; s2 < 13; s2 ++) do
        R2[1][s2] ← RD[s2]
      end for
      for (s3 = 0; s3 < 13; s3 ++) do
        R2[2][s3] ← RIt[s3]
      end for
      for (s4 = 0; s4 < 4; s4 ++) do
        R2[3][s4] ← RRa[s4]
      end for
```

Algorithm A6. *Cont.*

```
    /*Calculate fuzzy comprehensive evaluation index vector B[ ]*/
        for ($num$ = 0; $num$ < 13; $num$ ++) do
            for ($i$ = 0; $I$ < 13; $i$ ++) do
                for ($j$ = 0; $j$ < 4; $j$ ++) do
                    B[$num$] $\leftarrow$B[$num$]+W[$j$]*R2[$i$][$j$]
                end for
            end for
        end for
    Take y[ ] corresponding to the element with largest value in B[ ] as the scheme [$i$]
    end if
end for
```

Algorithm A7. acquisition of optimal machining element sequence

```
Input: all the MEs sequence TS[$i$][ ]
Output: optimal sequence OTS[ ]
$max \leftarrow 0$, $op \leftarrow 0$, $F$[ ] $\leftarrow$ 0
for ($i$ = 0; $i$ < n; $i$ ++) do//n is the number of sequences
    $diff\_m \leftarrow 0$, $diff\_t \leftarrow 0$, $diff\_D \leftarrow 0$
    for ($j$ = 0; $j$ < m−1; $j$ ++) do//m is the number of machining element
        $num \leftarrow$ TS[$i$][$j$]
        if (me[$num$].m_id $\neq$ me[$num$+1].m_id) then $diff\_m$++ end if
         if (me[$num$].m_id $\neq$ me[$num$+1].m_id | | me[$num$].t_id $\neq$ me[$num$+1].t_id) then
            $diff\_t$++
         end if
         if (me[$num$].$m\_id$ $\neq$ me[$num$+1].$m\_id$ | | me[$num$].$TAD$$\neq$ me[$num$+1].$TAD$) then
            $diff\_D$++
         end if
        end for
        $F$[$i$] $\leftarrow$ 0.53* diff_m+0.163* diff_t+0.297* diff_D
end for
for ($s$ = 0; $s$ < n; $s$ ++) do
    if (F[$s$]> max) then
        $max \leftarrow F$[$s$]
        $op \leftarrow s$
    end if
end for
for ($k$ = 0; $k$ < n; $k$ ++) do
    OTS[$k$] $\leftarrow$ TS[$op$][$k$]
end for
```

References

1. Al-wswasi, M.; Ivanov, A.; Makatsoris, H. A survey on smart automated computer-aided process planning (ACAPP) techniques. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 809–832. [CrossRef]
2. Kyprianou, L.K. Shape Classification in Computer-Aided Design. Ph.D. Thesis, Cambridge University, Cambridge, UK, 1980.
3. Han, J.H.; Pratt, M.; Regli, W.C. Manufacturing feature recognition from solid models: A status report. *IEEE Trans. Robot. Autom.* **2000**, *16*, 782–796. [CrossRef]
4. Joshi, S.; Chang, T.C. Graph-based heuristics for recognition of machined features from a 3D solid model. *Comput. Aided. Des.* **1988**, *20*, 58–66. [CrossRef]
5. Li, Y.G.; Ding, Y.F.; Mou, W.P.; Guo, H. Feature recognition technology for aircraft structural parts based on a holistic attribute adjacency graph. *Proc. Inst. Mech. Eng. B J. Eng. Manuf.* **2010**, *224*, 271–278. [CrossRef]
6. Marefat, M.; Kashyap, R.L. Geometric reasoning for recognition of three-dimensional object features. *IEEE Trans. Pattern. Anal. Mach. Intell.* **1990**, *12*, 949–965. [CrossRef]
7. Armstrong, G.T. A Study of Automatic Generation of Non-Invasive NC Machine Paths from Geometric Models. Ph.D. Thesis, University of Leeds, Leeds, UK, 1982.
8. Verma, A.K.; Rajotia, S. A review of machining feature recognition methodologies. *Int. J. Comput. Integr. Manuf.* **2010**, *23*, 353–368. [CrossRef]

9. Jüttler, B.; Kapl, M.; Nguyen, D.M.; Pan, Q.; Pauley, M. Isogeometric segmentation: The case of contractible solids without non-convex edges. *Comput. Aided. Des.* **2014**, *57*, 74–90. [CrossRef]
10. Kim, B.C.; Mun, D. Feature-based simplification of boundary representation models using sequential iterative volume decomposition. *Comput. Graph.* **2014**, *38*, 97–107. [CrossRef]
11. Tao, S.Q.; Huang, Z.D.; Ma, L.J.; Guo, S.S.; Wang, S.T.; Xie, Y.B. Partial retrieval of CAD models based on local surface region decomposition. *Comput. Aided. Des.* **2013**, *45*, 1239–1252. [CrossRef]
12. Campana, G.; Mele, M. An application to Stereolithography of a feature recognition algorithm for manufacturability evaluation. *J. Intell. Manuf.* **2020**, *31*, 199–214. [CrossRef]
13. Vandenbrande, J.H.; Requicha, A.A.G. Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Trans. Pattern. Anal. Mach. Intell.* **1993**, *15*, 1269–1285. [CrossRef]
14. Sunil, V.B.; Agarwal, R.; Pande, S.S. An approach to recognize interacting features from B-Rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique. *Comput. Ind.* **2010**, *61*, 686–701. [CrossRef]
15. Li, Y.; Wang, W.; Liu, X.; Ma, Y. Definition and recognition of rib features in aircraft structural part. *Int. J. Comput. Integr. Manuf.* **2014**, *27*, 1–19. [CrossRef]
16. Gao, S.; Shah, J.J. Automatic recognition of interacting machining features based on minimal condition subgraph. *Comput. Aided. Des.* **1998**, *30*, 727–739. [CrossRef]
17. Alain-Jérôme, F.; Egon, O. Intelligent agents for feature modelling in computer aided design. *J. Comput. Des. Eng.* **2018**, *5*, 19–40.
18. Liu, X.M.; Jia, Y.Q.; Chen, Z.R. Recognition of machining feature for engine cylinder blocks. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 1197–1204. [CrossRef]
19. Zhang, Y.; Dong, X.Y.; Li, D.S.; Zeng, Q.F.; Yang, S.H.; Gong, Y.D. Method for STEP–NC manufacturing feature recognition based on STEP and improved neural network. *Acta Aeronaut. Astronaut. Sin.* **2019**, *40*, 256–267. [CrossRef]
20. Duan, X.Y.; Yu, S.; Peng, F.Y.; Zheng, Y.; Xiang, F. Feature Matrix Based Complex Feature Hierarchical Recognition for Blank Model of Large Cabin Component. *J. Mech. Eng.* **2021**, *57*, 166–176. [CrossRef]
21. Luo, Z.Y.; Sheng, B.Y.; Chen, Y.K. Machining feature recognition of injection mold based on graph matching. *Mod. Manuf. Eng.* **2020**, *11*, 61–67. [CrossRef]
22. Berenji, H.R.; Khoshnevis, B. Use of artificial intelligence in automated process planning. *Comput. Mech. Eng.* **1986**, *5*, 47–55.
23. Huang, W.J. Fundamental Research on the Optimization of Process Planning for the Complicated Components. Ph.D. Thesis, School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Wuhan, China, 2012. [CrossRef]
24. Ma, S.M.; Liu, C.J.; Sun, Y.; Li, A.P. Research on decision-making of part feature machining program based on fuzzy reasoning. *J. Mach. Des.* **2014**, *31*, 15–19. [CrossRef]
25. Hu, Q.W.; Liu, D.L.; Du, P. Feature Processing Scheme Decision Based on Tsukamoto Fuzzy Reasoning. *Aerosp. Manuf. Technol.* **2018**, *4*, 5–9. [CrossRef]
26. Zhong, J.W.; Deng, Z.H.; Shi, Y.D.; Liu, W. Process Planning of Groove Feature Milling and Optimal Selecting for Tool Based on BP Neural Network. *Mach. Build. Autom.* **2019**, *48*, 51–55. [CrossRef]
27. Zhao, Y.D.; Tan, C.; Xu, J.C. Decision-making of processing links based on the fuzzy back-propagation network. Manufacturing Automation. *Mach. Build. Autom.* **2012**, *34*, 31–33. [CrossRef]
28. Guan, X.S.; Wang, Y.Q.; Tao, L.Y. Machining scheme selection of digital manufacturing based on genetic algorithm and AHP. *J. Intell. Manuf.* **2009**, *20*, 661–669. [CrossRef]
29. Hu, Y.J.; Wang, Y.; Wang, Z.L.; Wang, Y.Q.; Zhang, B.C. Machining scheme selection based on a new discrete particle swarm optimization and analytic hierarchy process. *Artif. Intell. Eng. Des. Anal. Manuf.* **2014**, *28*, 71–78. [CrossRef]
30. Deng, Z.H.; Lv, L.S.; Huang, W.L.; Shi, Y.D. A high efficiency and low carbon oriented machining process route optimization model and its application. *Int. J. Precis. Eng. Manuf. Technol.* **2019**, *6*, 23–41. [CrossRef]
31. Jiang, Z.G.; Jiang, Y.; Wang, Y.; Zhang, H.; Cao, H.J.; Tian, G.D. A hybrid approach of rough set and case-based reasoning to remanufacturing process planning. *J. Intell. Manuf.* **2019**, *30*, 19–32. [CrossRef]
32. Chen, Y.F.; Huang, Z.D.; Chen, L.P.; Wang, Q.F. Parametric process planning based on feature parameters of parts. *Int. J. Adv. Manuf. Technol.* **2006**, *28*, 727–736. [CrossRef]
33. Peng, H.; Wang, H.; Chen, D.J. Optimization of remanufacturing process routes oriented toward eco-efficiency. *Front. Mech. Eng.* **2019**, *14*, 422–433. [CrossRef]
34. Rao Krishna, A.G.; Rao, K.M. Optimisation of operations sequence in CAPP using an ant colony algorithm. *Int. J. Adv. Manuf. Technol.* **2006**, *29*, 159–164. [CrossRef]
35. Li, L.; Fuh, J.Y.H.; Zhang, Y.F.; Nee, A.Y.C. Application of genetic algorithm to computer-aided process planning in preliminary and detailed planning. *Robot. Comput. Integr. Manuf.* **2005**, *21*, 568–578. [CrossRef]
36. Zhou, G.H.; Tian, C.L.; Zhang, J.J.; Chang, F.T.; Lu, Q. Multi-objective process route optimization considering carbon emissions. *Int. J. Adv. Manuf. Technol.* **2018**, *96*, 1195–1213. [CrossRef]
37. Chen, B.; Jing, B.X. Process route optimization based on bacteria foraging and ant colony algorithm. *Chin. J. Eng. Des.* **2020**, *27*, 600–607, 624. [CrossRef]
38. An, X.H.; Chen, T. Optimization of process route based on intuitionistic fuzzy number and multi - objective optimization algorithm. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 1180–1191. [CrossRef]

39. Guo, X.Y.; Wang, L.; Zhang, Y.J. Machining Process Planning Based on Immune Genetic Algorithm under Rule Constraints. *China Mech. Eng.* **2020**, *31*, 482–488. [CrossRef]
40. Dou, J.P.; Li, J.; Su, C. Optimization of operation sequencing based on feasible operation sequence oriented genetic algorithm. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 1981–1990. [CrossRef]
41. Ouyang, H.B.; Shen, B. STEP-NC oriented process planning optimization based on hybrid genetic algorithm. *Comput. Integr. Manuf. Syst.* **2012**, *18*, 66–75. [CrossRef]
42. Liu, X.M.; Meng, F.F.; Li, A.P.; Gu, Z.Y. Research on Machining Step Sequencing of Machining Center Based on Polychromatic Sets Theory and Genetic Algorithm. *China Mech. Eng.* **2013**, *24*, 2437–2442. [CrossRef]
43. Zhou, X. Research on MBD Based 3D CAD Model Feature Recognition and Similarity Evaluation. Master's Thesis, Department of Mechanical and Engineering Electronic, Shandong University of Science and Technology, Qingdao, China, 2021.
44. Venu, B.; Komma, V.R.; Srivastava, D. STEP-based feature recognition system for B-spline surface features. *Int. J. Autom. Comput.* **2018**, *15*, 500–512. [CrossRef]
45. Ding, S.H.; Feng, Q.; Sun, Z.Y.; Ma, F. MBD Based 3D CAD Model Automatic Feature Recognition and Similarity Evaluation. *IEEE Access* **2021**, *9*, 150403–150425. [CrossRef]
46. Cheng, B.; Ding, J.L. Interacting Feature recognition based on Hint and Feature Main Base Face. *Mach. Tool. Hydraul.* **2019**, *47*, 72–77. [CrossRef]
47. Zhou, Y.P.; Zhang, J.; Qin, W. Feature recognition methodology for large thin-wall part of spacecraft. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 2505–2512. [CrossRef]
48. Guo, F.L.; Sun, G.Z.; Zhou, J.T.; Shao, L. Machining Feature Recognition Based on Graph Twice Decomposition. *Mach. Des. Manuf.* **2013**, *5*, 56–59. [CrossRef]
49. Wang, B. Research on Machining Feature Recognition and Process Path Optimization Based on MBD Model. Master's Thesis, Department of Mechanical and Engineering Electronic, Shandong University of Science and Technology, Qingdao, China, 2022.
50. Wei, Z.Y. Application Research of Fuzzy Comprehensive Evaluation in the Optimization of Feasibility Study Scheme. Master's Thesis, Department of Civil Engineering, Lanzhou Jiaotong University, Lanzhou, China, 2020.
51. Zhou, Y.F. Study of the Key Technologies of Hybrid CAPP System Based on the Knowledge-Driven. Master's Thesis, Department of Electronic Engineering, North University of China, Taiyuan, China, 2012.