Similarity Measurement of the Geometry Variation Sequence of Intermediate Process Model

Chunlei Li (602lcl-602lcl@163.com)
Baoji University of Arts and Sciences https://orcid.org/0000-0002-7205-6885
Liang Li
Baoji University of Arts and Sciences
Xiaoye Wang
Baoji University of Arts and Sciences

Original Article

Keywords: Process Similarity Measurement; Intermediate Process Model; Geometry Change; Geometry Variation Sequence

Posted Date: May 5th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-25784/v1

Similarity Measurement of the Geometry Variation Sequence of Intermediate Process Model

Chunlei Li[1,2], Liang Li[1,2], Xiaoye Wang[1,2]

(1. School of mechanical engineering, Baoji University of Arts and Sciences, Baoji Shaanxi, 721016, P.R. China

2. Shaanxi Key Laboratory of Advanced Manufacturing and Evaluation of Robot Key Components, Baoji Shaanxi, 721016, P.R. China)

Abstract:

The reuse of machining process, by which the process for a new mechanical part is determined by referencing to the existing and matured processes, is an effective way of improving manufacturing and supporting innovation. To conduct the effective reuse it is necessary to express and retrieve a specific process. A kernel technique of the expression and the retrieval is to measure the similarity of part's geometry variation during the machining. To address this problem, a general framework of measuring the similarity between parts is proposed in this work. The geometry variation sequence of intermediate process model is established, and then a method to measure its similarity is invented. Two case studies are rendered and the result reveals that the proposed method is effective and can provide the support for the process retrieval and reuse in industry.

Key words： Process Similarity Measurement; Intermediate Process Model; Geometry Change; Geometry Variation Sequence

1 Introduction

The manufacturing of mechanical parts is undergoing a major strategic shift from "automation" to "intelligence", and the later becomes nowadays a popular topic. To pursue the intelligent manufacturing, it is important to design a manufacturing process that matches the products' design and capacities of a specific manufacturing enterprise. Being able to retrieve and reuse the existed process case reasonably, is one of the most effective ways to improve manufacturing and support innovation.

Many researches have focused on the process retrieval and reuse. Traditionally, research on evaluating the similarity between process instances is based on three-dimensional (3-D) model retrieval. Chang et al [1] built up an individual index for each part in the case base, and this index contained geometric shape information of the part. A part with geometrically similar shape can be efficiently searched by using the index from the similar cases. El-Mehalawi et al [2] used an Attributed Adjacency Graph (AAG) method to represent a 3-D model, and then the AAG is further stored and measured, in order to conduct process similarity retrieval and reuse. Cuillière et al [3] applied a vector method to represent 3-D model to automatically compare vectors. Zhang et al [4] developed a system for the indexing and retrieval of 3D models. In the system, some features, such as the cord histogram and the 3-D shape spectrum, are utilized to calculate the similarity between 3-D models. The active learning method is used to improve the annotation efficiency. Hou et al [5] proposed a semi-supervised semantic clustering method based on Support Vector Machines (SVM), to organize the 3D models semantically. An unified search strategy is

Email:602lcl-602lcl@163.com

employed, in which semantic constraints are applied to the retrieval by using the resulting clusters. Ohbuchi et al [6] explored a method to improve feature distance computation by employing unsupervised learning of the subspace of 3-D shape features from a corpus. Among above researches, the similarity of 3-D models was the main basis for process searching. Unfortunately, the geometry change of part model during the manufacturing processes is ignored. In fact, the geometry change caused by machining operation is closer to the nature of manufacturing. Besides the 3-D model retrieval and machining operation sequence retrieval are also an available method of process retrieval and reuse. Biundo et al [7] provided a process planning system based on deductive reasoning mechanism. The system consists of the deductive reasoning module and the planning reuse module. Kambhampati [8] focused on the selection of the reusable process from more alternative planning schemes, and developed a method to calculate the distance between alternative planning schemes and target object. Liu et al [9] established a mathematical model of machining process route, and then measured the similarity between two process routes by the Manhattan distance. Jiang et al [10] discribed a hybrid method combing rough set and case-based reasoning for re-manufacturing process planning, where the case-based reasoning is employed for similarity calculation to effectively identify the most suitable solution from a case database. In these machining operation retrieval methods, although the similar operation planning scheme could be retrieved from the case base, the geometry change of part model caused by machining operation had not been analyzed. Meanwhile one machining operation might corresponds to various forms of geometry change, but their process implementations could be not quite similar. To summarize, existing methods are lack of consideration for the geometric change of part model. This can cause decreases in the suitability and effectiveness in reusing the manufacture instances.

Here in this research, a novel method of similarity measurement is proposed in order to provide better support for process instance retrieval. The proposed methodology contains the following three parts:

(1) The geometry change of part model caused by machining operation is extracted, and described using the attributed adjacency graph. Based on graphical representation of geometry change, the graph model is transformed into a unique string.

(2) After all machining operations in process are represented by strings, the geometry variation sequence was built up with all the geometry changes in processing sequence.

(3) A method to measure similarity of the geometry variation sequence is developed to evaluate the process similarity of parts.

Because the similarity of process instances is evaluated based on the geometry change of intermediate process models, the machining process retrieved from case base is more consistent with the target object. As a result, the process reuse is improved.

2 General framework of similarity measurement of geometry variation sequence

In the modeling of the geometry variation of intermediate process models, the geometry change between the former and the current procedures needs to be extracted and expressed. Then the whole intermediate process models in a machining example are analyzed identically, and all the geometry changes extracted from intermediate process models constitute the geometry

variation sequence in processing sequence. After obtaining the geometry variation sequence, the process similarity of parts can be obtained by measuring the similarity of geometry variation sequences. The general framework of similarity measurement of geometry variation sequence is given in Fig. 1. The intermediate process models parsed by process instance are the input, and the geometry variation sequence is the objective for process similarity measurement.

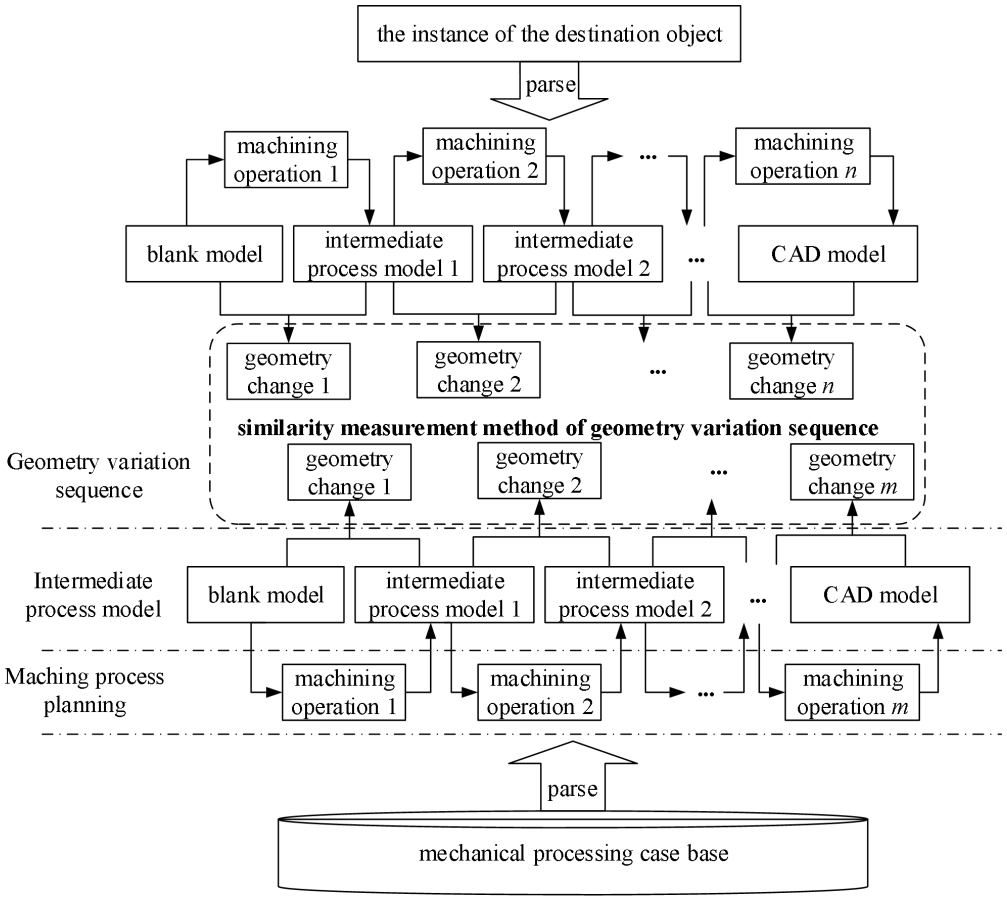

Fig. 1 General framework of similarity measurement of geometry variation sequence

3 Acquisition and representation of geometry change

3.1 Acquisition of geometry change

In machining, the part is gradually changed from a prismatic billet to the design model. A machining operation in the process corresponds to a reduction of material, which can be perceived as a geometry change of 3-D model. Based on the relation between machining operation and geometry change, the geometry change caused by the ith machining operation op_i is defined as follows.

$$GV_i = \sum_{j=1}^{n_i} g_i^j = IPM_{i-1} - IPM_i , \tag{1}$$

where GV_i represents the geometry change corresponding to op_i, g_i^j represents the material removed by the jth sub-operation in op_i, IPM_i represents the intermediate process model generated by op_i, n_i is the number of sub-operations in op_i. Obviously, IPM_0 represents the initial billet model. IPM can be generated by two ways: a) the IPMs are generated manually by using the

model based definition (MBD) technique [11-12]; or b) they can be automatically computed. In the first way, the human-machine interaction is involved and its efficiency is low. The automatic generation techniques of intermediate process model described in [13-14] are adopted in this work.

As revealed in Equ. 1, the geometry change can be regarded as the result of boolean subtraction between the former and the current *IPMs*. A common mechanical part, the transmission shaft, is provided as an example shown in Fig. 2. The geometry change caused by the first machining operation, rough turning, is shown in Fig. 3.

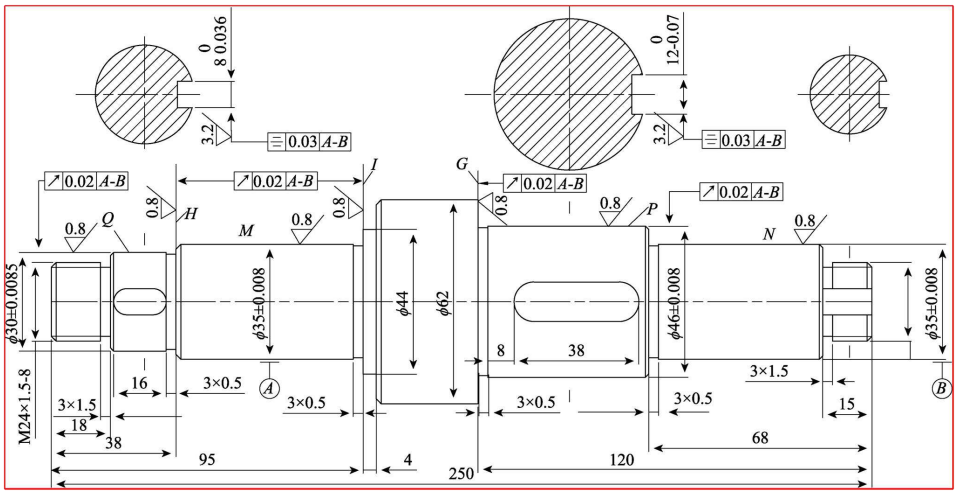

Fig. 2 A transmission shaft part

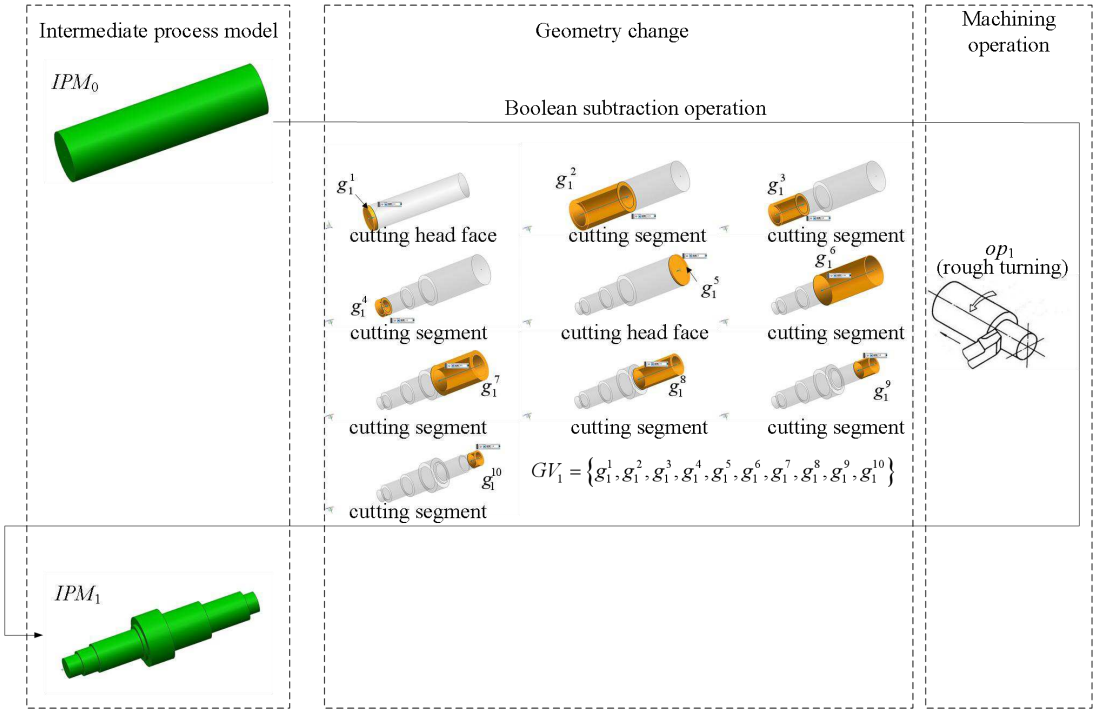

Fig. 3 Extraction of geometry change

3.2 Graphical representation of geometry change

As seen in Fig. 3, the geometry change can be treated as a collection of 3-D solids, so the

representation of geometry change is transformed into the representation of the corresponding 3-D solid. Generally, representing the 3-D solid by using AAG is a widely adopted approach. Thus, the graphical representation of 3-D solid is defined as follows.

$$AAG=(V,\ E,\ VAS,\ EA), \tag{2}$$

where V is the collection of vertexes in AAG, and the vertex in AAG represents the surface of the 3-D solid. For each surface of solid model, there is an existing vertex v_i corresponding to it. E is the collection of edges in AAG, and the edge in AAG represents the adjacency relation between the surfaces of solid model. If there are two existing adjacent surfaces v_i and v_j, there is an existing edge $e_{ij} \in E$. VAS is the attribute collection of the vertex in AAG, including the connectivity of the surface, the type of the surface and the area of the surface. EA is the type attribute of the edge in AAG. If e_{ij} is a straight line, its corresponding attribute value is 0, if e_{ij} is a plane curve, its corresponding attribute value is 1, Space curve corresponds to 2.

The graphical representation model of geometry change is constructed the following three steps.

Step 1: Based on section 3.1, the geometry change GV_i between IPM_{i-1} and IPM_i is extracted.

Step 2: by traversing all 3-D solids in GV_i, the AAG (V, E, VAS, EA) for each 3-D solid is created.

Step 3: When all 3-D solids in GV_i have been expressed as their corresponding AAG, GV_i can be expressed as a collection of AAG that $GV_i=\{g_i^j \mid j=1, 2, \ldots, n_i\}=\{AAG_i^j \mid j=1, 2, \ldots, n_i\}$, where AAG_i^j is graphical representation of the 3-D solid g_i^j.

For the geometry change GV_1 shown in Fig. 3, its graphical representation is shown in Fig. 4.

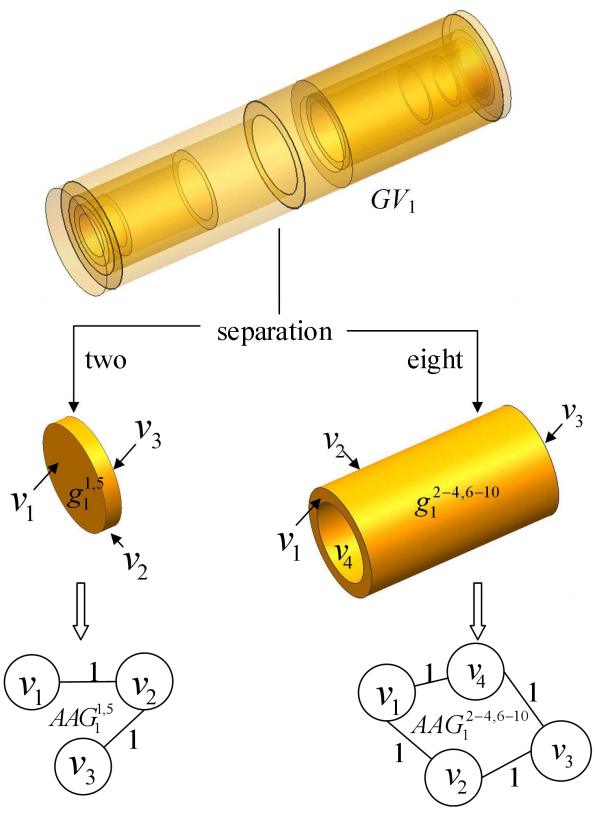

Fig. 4 Graphical representation of geometry change

3.3 String representation of geometry change

The Mathematical representation of geometry change is the foreshadowing of subsequent similarity calculation, but the graph representation model is not easy to be stored and analyzed. Therefore, a method of transforming the graph representation model into a unique string is provided here. Compared to the graph representation, the string is easy to be stored, and the efficiency in similarity calculation can be improved dramatically.

The transforming method is as follows:

Step 1: To reorder the vertexes in *AAG*, three sub-steps is designed as following.

Step 1.1: The vertexes in *AAG* is ordered by the connective number (or connectivity). If the number of neighbor faces of v_i is greater than the number of neighbor faces of v_j, v_i should be arranged prior to v_j.

Step 1.2: For those vertexes whom connectivity are same, they should be ordered by the occurrence probability of the type of the surface. Assuming that the connectivity of v_i and v_j are equal, if the occurrence probability of the type of v_i is greater than that of v_j, v_i should be arranged prior to v_j. For the 3-D models, the occurrence probability of plane is assumed greater than that of cylindrical surface, the cylindrical surface is greater than the cone surface, and the cone surface is greater than the other surface.

Step 1.3: For those vertexes whose connectivity and types are same, they are ordered by their areas. Assuming that the connectivity and types of v_i and v_j are both equal, if the area of v_i is greater than that of v_j, v_i should be arranged prior to v_j.

Based on step 1, the collection of vertexes $V=\{v_1, v_2, \ldots\}$is reordered, and a sorted collection of vertexes is represented by $Ord(V)=\{v'_1, v'_2, ...\}$, where v'_i represents the sorted location of the vertex.

Step 2: The relation set of sorted vertexes is defined as $Vere(v'_i, v'_j)=\{VA(v'_i), EA(e'_{ij}), VA(v'_j)\}$, where $VA(v'_i)$ and $VA(v'_j)$ denote the types of the sorted vertexes v'_i and v'_j, respectively. In this work, the plane is represented as the string P_l, the cylindrical surface C_y, cone surface C_i, and the flank of thread T_h. e'_{ij} represents the adjacent edge between v'_i and v'_j, and $EA(e'_{ij})$ represents the type attribute value of e'_{ij}.

Step 3: All adjacent vertexes in $Ord(V)$ are evaluated in orders, and then a new string $Str=\{Vere(v'_1, v'_2), Vere(v'_2, v'_3), ..., Vere(v'_i, v'_{i+1}), ...\}$ is obtained to represent the *AAG*.

The *AAG* of the example shown in Fig. 4 is ordered and its vertexes are plotted in Fig. 5. As seen in Fig. 5, AAG_1^1 and AAG_1^5 are expressed as a unique string $C_y1P_lP_l\_P_l$. AAG_1^3, AAG_1^4, *AAG*, AAG_1^7, AAG_1^8, AAG_1^9 and AAG_1^{10} are expressed as a unique string $P_l\_P_lP_l1C_yC_y\_C_y$. After all 3-D solids in GV_1 are expressed as their corresponding strings, GV_1 is transformed into a string collection that includes two strings, i.e., $C_y1P_lP_l\_P_l$ and $P_l\_P_lP_l1C_yC_y\_C_y$.

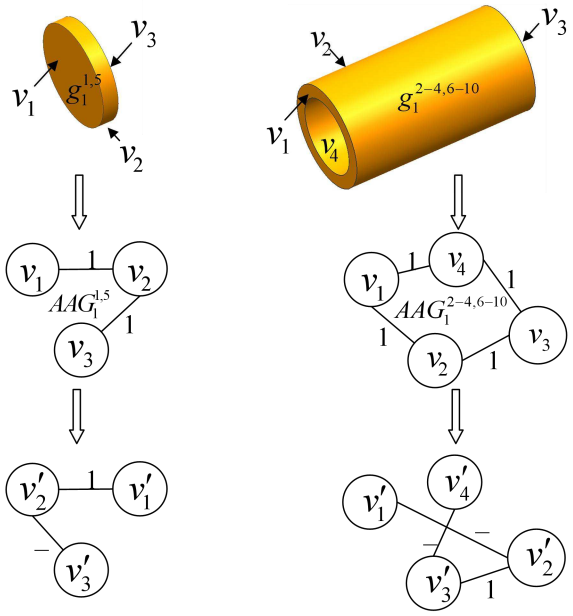

Fig. 5 Attribute Adjacent Graph with sorted vertexes

4 Establishment of geometry variation sequence

Suppose the machining process of a part is $OL(op_1 \rightarrow op_2 \rightarrow \cdots \rightarrow op_n)$, the geometry change caused by machining operation op_i(i=1, 2, …, n) can be showed as the collection of 3-D solids based on section 3.1. The equation is

$$GV_i = \{g_i^j \,|\, j = 1, 2, ..., n_i\}. \tag{3}$$

Then GV_i can be expressed as a collection of *AAG*s based on section 3.2, by the following equation:

$$GV_i = \{AAG_i^j \,|\, j = 1, 2, ..., n_i\}, \tag{4}$$

where AAG_i^j is the graphical representation of g_i^j. On this basis, AAG_i^j can be transformed into a string collection based on section 3.3, as shown in Equ. 5.

$$GV_i = \{Str_i^j \,|\, j = 1, 2, ..., n_i\}, \tag{5}$$

where Str_i^j is the string representation of AAG_i^j.

After all machining operations in *OL* have been expressed as their corresponding strings, the geometry variation sequence (*GVS*) of *OL* can be constructed as follows.

$$\begin{aligned} GVS &= \{GV_1, GV_2, ..., GV_n\} \\ &= \{\{Str_1^j \,|\, j = 1, 2, ..., n_1\}, ..., \{Str_n^j \,|\, j = 1, 2, ..., n_n\}\} \end{aligned}. \tag{6}$$

The geometry variation sequence created in this way can fully and precisely describe geometric evolution situation of all *IPM*s. However, some machining operations do not cause significant topological change of 3-D model. For example, finishing and grinding are adopted for the requirement of dimensional precision, and these operations lead not to an obvious change of part model. Thus, these geometry changes are eliminated in the construction of *GVS*.

5 Similarity measurement of the geometry variation sequences

5.1 Similarity computation of 3-D solid

According to section 3.3, 3-D solids can be transformed into unique strings, so the similarity computation of 3-D solid is translated into the similarity computation of strings. Traditionally, research on similarity computation of string is based on Levenshtein distance, the counting process of this method is as follows.

Step 1: The matching relation matrix **Ld** ($(|Str_1|+1)\times(|Str_2|+1)$) between strings Str_1 and Str_2 is established, where $|Str_1|$ represents the length of Str_1, $|Str_2|$ represents the length of Str_2.

Step 2: All the elements in **Ld** are recursively calculated according to the following equation.

$$\mathbf{Ld}(x,y)=\begin{cases} x & (y=0) \\ y & (x=0) \\ \min(\mathbf{Ld}(x\text{-}1,y)+1,\mathbf{Ld}(x,y\text{-}1)+1,\mathbf{Ld}(x\text{-}1,y\text{-}1)+d_{xy}) & (0<x\le|Str_1|,0<y\le|Str_2|) \end{cases}, \tag{7}$$

where d_{xy} represents the editing times. If the x^{th} character in Str_1 and the y^{th} character in Str_2 are equal, then d_{xy}=0, otherwise d_{xy}=1.

Step 3: The lower-right element $\mathbf{Ld}(|Str_1|,|Str_2|)$ is selected as the levenshtein distance ld between Str_1 and Str_2.

Setp 4: The similarity is calculated based on equations 8 or 9.

$$sm_{str1}(Str_1,Str_2)=1-\frac{ld}{|Str_1|+|Str_2|} \tag{8}$$

$$sm_{str2}(Str_1,Str_2)=1-\frac{ld}{\max(|Str_1|,|Str_2|)} \tag{9}$$

The above counting process is the classical similarity measurement method based on Levenshtein distance. However, this method takes only the editing times into consideration, the impact of the common substring of two strings on the similarity is ignored. As a result, the suitability of classical Levenshtein method is low. Suppose there are three strings: P_lP_l1, C_y1P_l and $C_y\_C_y$. Although the lengths of the three strings are all 3, P_lP_l1 and C_y1P_l have substring P_l or 1 in common, while P_lP_l1 and $C_y\_C_y$ have no substring in common. Based on intuitive experience, the similarity between P_lP_l1 and C_y1P_l should be greater than that between P_lP_l1 and $C_y\_C_y$. Nevertheless, the similarities obtained using equation 8 and 9 are

$$sm_{str1}(P_lP_l1,\ C_y1P_l)=0.5$$

$$sm_{str1}(P_lP_l1,\ C_y\_C_y)=0.5$$

$$sm_{str2}(\ P_lP_l1,\ C_y1P_l)=0$$

$$sm_{str2}(\ P_lP_l1,\ C_y\_C_y)=0,$$

which can not reflect the actual situation. Therefore, Levenshtein method is improved in this work. The improved Levenshtein distance method is different in an important aspect from classical method. In addition to the consideration of the editing times, the common substring of two strings is also considered as an important factor for the similarity computation. The improved equation replacing Equations 8 and 9 is as follows.

$$sm_{str}(Str_1,Str_2)=\frac{lcs}{ld+lcs+\dfrac{|Str_1|+|Str_2|-2\kappa}{|Str_1|+|Str_2|}}, \tag{10}$$

where lcs represents the length of common substring of Str_1 and Str_2, κ represents the length of common prefix of Str_1 and Str_2. Based on the Equ. 10, the length of common substring of P_lP_l1 and C_y1P_l is 1, the length of common substring of P_lP_l1 and $C_y\_C_y$ is 0. Meanwhile, P_lP_l1 and

C_y1P_l have no prefix in common, and P_lP_l1 and $C_y\_C_y$ nither. Then the similarity of strings by using equation 10 is obtained as follows.

$$sm_{str}(P_lP_l1, C_y1P_l)=0.2$$

$$sm_{str}(P_lP_l1, C_y\_C_y)=0,$$

which is more consistent with actual situation.

In order to demonstrate the validity of the proposed method, two 3-D solids shown in Fig. 5 are evaluated. Based on section 3.3, the cylinder is translated into the unique string $C_y1P_lP_l\_P_l$ and the ring-shaped solid is translated into the unique string $P_l\_P_lP_l1C_yC_y\_C_y$. Then **Ld**(7×10) between $C_y1P_lP_l\_P_l$ and $P_l\_P_lP_l1C_yC_y\_C_y$ is calculated based on equation 7, as shown in Fig. 6.

	P_l	_	P_l	P_l	1	C_y	C_y	_	C_y
	0	1	2	3	4	5	6	7	8	9
C_y	1	1	2	3	4	5	5	6	7	8
1	2	2	2	3	4	4	5	6	7	8
P_l	3	2	3	2	3	4	5	6	7	8
P_l	4	3	3	3	2	3	4	5	6	7
_	5	4	3	4	3	3	4	5	5	6
P_l	6	5	4	3	4	4	4	5	6	6

Fig. 6 Matching relation matrix **Ld** between cylinder and ring solid

As seen in Fig. 6, **Ld** between $C_y1P_lP_l\_P_l$ and $P_l\_P_lP_l1C_yC_y\_C_y$ is 6. Meanwhile, $C_y1P_lP_l\_P_l$ and $P_l\_P_lP_l1C_yC_y\_C_y$ have no common prefix, then κ=0. They have substring $P_l\_P_l$ in common, resulting in *lcs*=3. The similarity between the cylinder and the ring-shaped solid is calculated by using Equ. 10, and the result is $sm_{str}(C_y1P_lP_l\_P_l, P_l\_P_lP_l1C_yC_y\_C_y)$=0.3.

5.2 Similarity computation of geometry changes

For two geometry changes, suppose their collection representations are GV={Str_1, Str_2,…, Str_p} and $GV' = \{Str_1', Str_2', ..., Str_q'\}$, where p and q denote the numbers of strings in GV and GV', respectively. Then a bipartite graph is constructed based on these two collections and the similarities among them, as shown in Fig. 7. In the figure, $sm_{str} = (Str_i, Str_j')$ represents the attribute value of edge in bipartite graph.

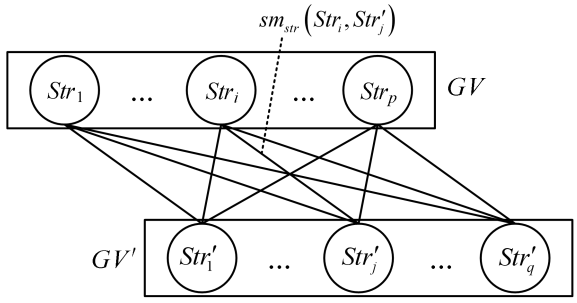

Fig. 7 Construction of the bipartite graph

The edge weight between Str_i and Str'_i is formulated as follows:

$$w_{ij} = \frac{2}{p+q} sm_{str}(Str_1, Str_2), \tag{11}$$

And the similarity computation of geometry change can be translated into the problem of solving the matching with maximum weighted sum in the bipartite graph, so the classical algorithm of Kuhn-Munkres [15-16] is applied to solve the problem.

For *GV* and *GV'*, if the numbers of strings contained by them are equal, the Kuhn-Munkres algorithm can be applied directly to solve the matching with maximum weighted summation, and the maximum weighted summation is the similarity between *GV* and *GV'*. When the number of strings in *GV* is not equal to the number of strings in *GV'*, say *q* is greater than *p*, all combinations with selecting *p* strings from *GV'* is analyzed, in order to find the matching with maximum weighted summation between each combination and *GV* by Kuhn-Munkres algorithm. Finally the maximum value from all maximum weighted summation is chosen as the similarity between *GV* and *GV'*.

The pseudo-code of calculating similarity of geometry change is shown in Fig. 8.

```
function t = gv_sim(gv1,gv2)
  len1=length(gv1);
  len2=length(gv2);
  T=[];
  if len1<=len2
  for i=1:len2
      c(i)=i;
  end
  M=nchoosek(c,len1);
  m=size(M,1);
  for u=1:m
       for i=1:len1
            for j=1:len1
                A(i,j)=(2/(len1+len2))*sm(gv1{i},gv2{M(u,j)});
            end
       end
       B=[];
       [B,T(u)]= Kuhn_Munkres(A,len1);
  end
  t=max(T);
  else
  for i=1:len1
       c(i)=i;
  end
  M=nchoosek(c,len2);
  m=size(M,1);
```

```
for u=1:m
    for i=1:len2
        for j=1:len2
            A(i,j)=(2/(len1+len2))*sm(gv1{M(u,i)},gv2{j});
        end
    end
    B=[];
    [B,T(u)]= Kuhn_Munkres(A,len2);
    end
    t=max(T);
end
end
```

Fig. 8 The pseudo-code of calculating similarity of geometry change

5.3 Similarity computation of geometry variation sequences

When calculating the similarity of geometry variation sequences, if the partial difference of two sequences is larger, the global similarity error may be larger. In order to avoid this disadvantage, a revision of Blast algorithm [17] is constructed. The revised method contains the following two steps: (1) Search for the optimum matching sequences of two geometry variation sequences, and (2) Calculate the similarity of the optimum matching sequences.

Given two geometry variation sequences: $GVS_1=\{GV_1, GV_2, \ldots, GV_{n1}\}$ and $GVS_2 = \{GV_1', GV_2', ..., GV_{n2}'\}$. The numbers of geometry changes in GVS_1 and GVS_2 are not equal sometimes; meanwhile more attention should to be paid to the similarity of the key geometric topological changes. So the key geometry variation sequences GVS_1^* needs to be extracted from GVS_1, and then the geometry changes from GVS_2, which is the most similar to GVS_1^*, is adopted to form the matching sequence GVS_2'. On this basis, the similarity between GVS_1 and GVS_2 can be represented by the similarity between GVS_1^* and GVS_2'. GVS_1, GVS_2 and GVS_1^* can be obtained based on section 4, GVS_2' is constructed as follows:

Step 1: Initialize i=1.

Step 2: Select the geometry change GV_j' from GVS_2 that is most similar to the ith geometry change GV_i^* in GVS_1^*, and then define a new generated $GV_i''=GV_j'$.

Step 3: Record the position of GV_j' in GVS_2, remove the geometry changes from the first geometry change to the jth geometry change in GVS_2, and then update $GVS_2 = \{GV_{j+1}', GV_{j+2}', ..., GV_{n2}'\}$.

Step 4: Implement $i=i+1$. If the number of the geometry changes in GVS_1^* is greater than i and the updated GVS_2 is nonempty, go to Step 2. Otherwise, select the new generated sequence $\{GV_1'', GV_2'', ...\}$ as GVS_2'.

The aforementioned algorithm is shown in Fig. 9.

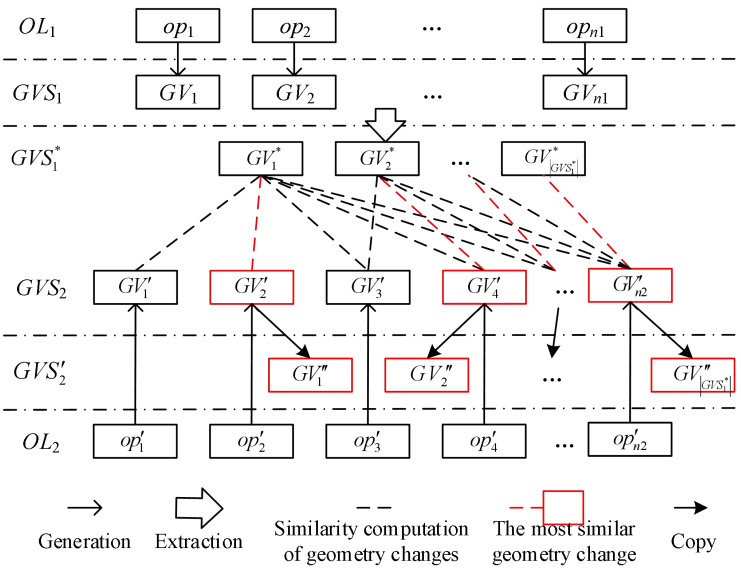

Fig. 9 Acquisition process of the matching sequence

After obtaining the matching sequence, the similarity between GVS_1 and GVS_2 is calculated based on equation 12.

$$Sim\left(GVS_1, SVS_2\right) = \frac{\left|GVS_2'\right|}{\left|GVS_1^*\right|}\sqrt{\frac{1}{\left|GVS_2'\right|}\sum_{i=1}^{\left|GVS_2'\right|} sm_{gv}\left(GV_i^*, GV_i''\right)}, \tag{12}$$

where $Sim(GVS_1, GVS_2)$ represents the similarity between GVS_1 and GVS_2, ; sm_{gv} represents the similarity of geometry changes based on section 5.2, $\left|GVS_1^*\right|$ represents the length of GVS_1^*, $\left|GVS_2'\right|$ represents the length of GVS_2' .

6 Case study

In this section, two cases are rendered to demonstrate the feasibility and effectiveness of the proposed method for similarity retrieval of machining process.

6.1 Case I

The part shown in Fig. 2, together with two parts shown in Fig. 10 and 11, are taken as examples. Obviously, two shaft parts in Fig. 2 and 10 are similar in shape, and the cover part in Fig. 11 is different with them.

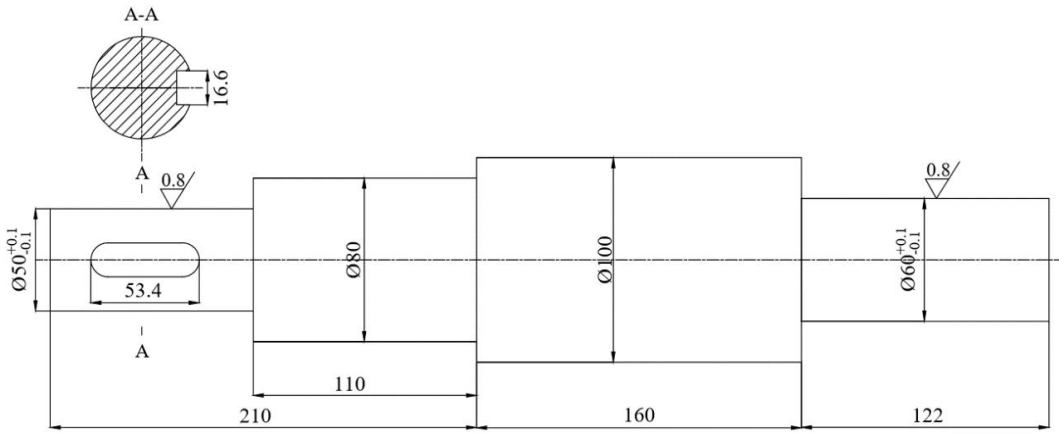

Fig. 10 A stepped shaft

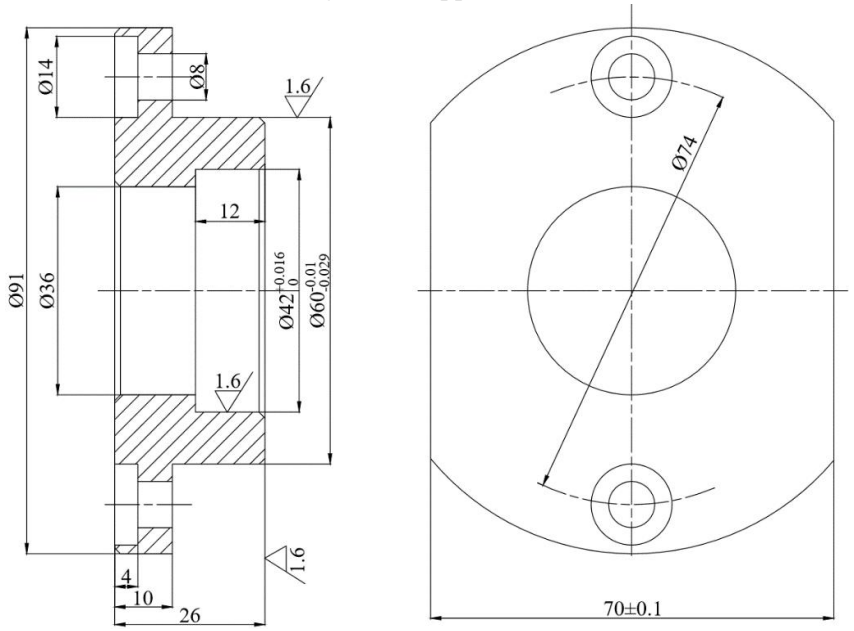

Fig. 11 A cover part

Based on section 4, the geometry variation sequences GVS_1, GVS_2 and GVS_3 that are corresponding to the three parts shown in Fig. 2, 10 and 11 respectively, can be constructed, and then GVS_1^* can be extracted from GVS_1 , as shown in Fig. 12.

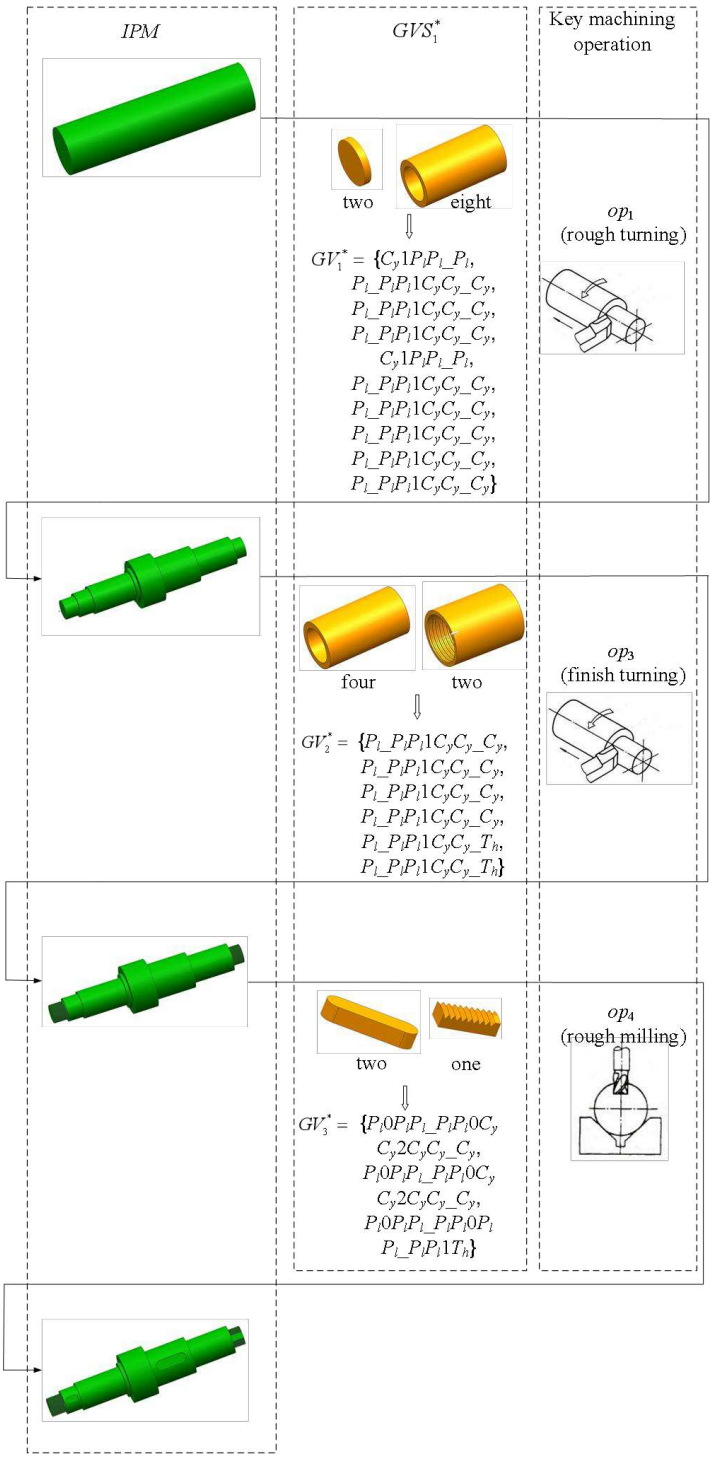

Fig. 12 The key geometry variation sequence GVS_1^*

Based on section 5.3, the geometry changes from GVS_2, which is similar to GVS_1^* the most, is chosen to form the matching sequence GVS_2', and the geometry changes from GVS_3, which is similar to GVS_1^* the most, is chosen to form the matching sequence GVS_3'. GVS_2' and GVS_3' are shown in Fig. 13 and 14, respectively.

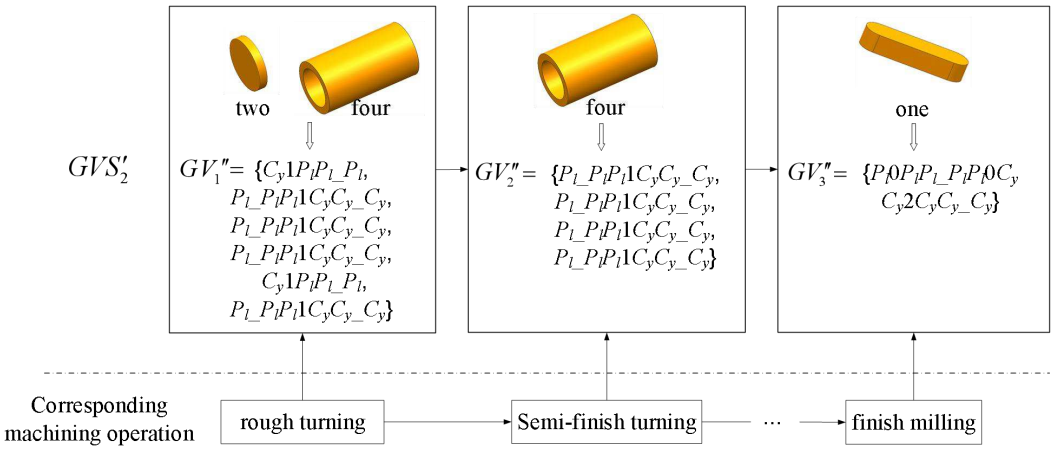

Fig. 13 The matching sequence GVS'_2

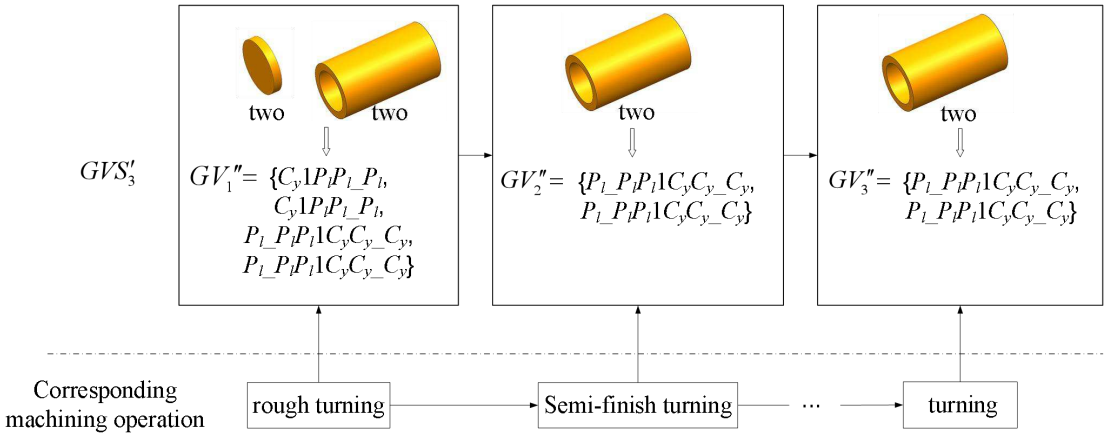

Fig. 14 The matching sequence GVS'_3

The similarities between GVS_1 and GVS_2, and GVS_1 and GVS_3, are evaluated using Equ. 12. The result is shown in Tab. 1.

Table 1. The matching sequence GVS'_3

i	$Sim(GVS_1, GVS_i)$
2	0.6958
3	0.4961

Out of our intuition, the similarity of similar parts should be greater than that of different types of parts. The result in Tab.1 is confirmed with the intuition. Meanwhil,e if two parts in the same type have some different geometric structures and corresponding evolution process, the similarity of the two parts obtained by measuring the geometry variation sequence is less than the similarity obtained by evaluating the machining operations. This makes the similarity measurement more detailed and comprehensive, which can provide more accurate matching results when the similar manufacturing process needs to be retrieved from the matured processes.

6.2 Case II

In the second case, the process base is obtained from a manufacturing enterprise as an example. Ten process in the base are selected at random. The geometry variation sequences of the

parts are constructed based on section 3 and 4, as shown in Tab. 2.

Table 2. The selected machining instances

Number	Part name	Machining operation route	Key geometry variation sequence
P_1	drive shaft	rough turning→semi-finish turning→finish turning→rough milling→semi-finish milling→finish milling→grinding	{two $C_y1P_lP_l\_P_l$, eight $P_l\_P_lP_l1C_yC_y\_C_y$ }→{four $P_l\_P_lP_l1C_yC_y\_C_y$, two $P_l\_P_lP_l1C_yC_y\_T_h$}→{two $P_l0P_lP_l\_P_lP_l0C_yC_y2C_yC_y\_C_y$, one $P_l0P_lP_l\_P_lP_l0P_lP_l\_P_lP_l1T_h$}
P_2	stepped shaft	rough turning→semi-finish turning→finish turning→milling	{two $C_y1P_l\,P_l\_P_l$, four $P_l\_P_lP_l1C_yC_y\_C_y$}→{one $P_l0P_lP_l\_P_lP_l0C_yC_y2C_yC_y\_C_y$}
P_3	cylinder liner	casting→rough turning→semi-finish turning→boring	{two $C_y1P_lP_l\_P_l$, two $P_l\_P_lP_l1C_yC_y\_C_y$}→{one $C_y1P_lP_l\_P_l$}
P_4	guide sleeve	rough turning→semi-finish turning→drilling→boring→ grinding	{two $C_y1P_lP_l\_P_l$, two $P_l\_P_lP_l1C_yC_y\_C_y$}→{two $C_y1P_l\,P_l\_P_l$}→{two $P_l\_P_lP_l\,1C_yC_y\_C_y$}
P_5	bearing Cover	rough turning→semi-finish turning→drilling→turning (inner circle)→milling →drilling	{two $C_y1P_lP_l\_P_l$, two $P_l\_P_lP_l1C_yC_y\_C_y$}→{two $C_y1P_lP_l\_P$}→{two $P_l\_P_lP_l1C_yC_y\_C_y$}→{two $P_l0C_yC_y1P_lP_l\_P_l$}→{four $C_y1P_lP_l\_P_l$}
P_6	flange	casting→rough turning→semi-finish turning→finish turning→drilling→rough turning(inner circle)→finish turning(inner circle) →drilling	{two $C_y1P_lP_l\_P_l$, one $P_l\_P_lP_l\,1C_yC_y\_C_y$}→{one $C_y1P_lP_l\_P$}→{one $P_l\_P_lP_l1C_yC_y\_C_y$}→{six $C_y1P_l\,P_l\_P_l$}
P_7	valve cover	casting→rough turning→semi-finish turning→finish turning→ boring(center hole)→drilling	{two $C_y1P_lP_l\_P_l$, three $P_l\_P_lP_l1C_yC_y\_C_y$}→{one $C_y1P_l\,P_l\_P_l$}→{four $C_y1P_l\,P_l\_P_l$}
P_8	close lid	casting→rough turning→semi-finish turning→milling→ drilling(reaming)	{one $P_l\_P_lP_l1C_yC_y\_C_y$}→{one $P_l\_P_lP_l\,1C_yC_y\_C_y$}→{eight $C_y1P_lP_l\_P_l$}
P_9	Triangular bearing box	casting→milling→milling→rough boring→finish boring→milling→drilling→ milling→drilling	{two $P_l\_P_lP_l\,0P_lP_l\_P_lP_l0P_lP_l\_P_l$}→{two $P_l\_P_lP_l\,0P_lP_l\_P_lP_l0P_lP_l\_P_l$}→{one $P_l\_P_lP_l1C_yC_y\_C_y$}→{two $P_l\_P_lP_l1C_yC_y\_C_y$}→{six $C_y1P_lP_l\_P_l$}→{two P_l

			$\_P_lP_l0P_lP_l\_P_lP_l0P_lP_l\_P_l\}\rightarrow${two $C_y1P_lP_l\_P_l\}$
P_{10}	valve body	casting→milling→milling→	{two $P_l\_P_lP_l$ $1C_yC_y\_C_y\}\rightarrow${one $P_l\_P_lP_l$
		milling→rough boring→finish	$1C_yC_y\_C_y\}\rightarrow${one $P_l\_P_lP_l1C_yC_y\_C_y\}\rightarrow${six $P_l\_P_lP_l$
		boring→rough	$1C_yC_y\_C_y\}\rightarrow${one $P_l\_P_lP_l1C_yC_y\_C_y\}\rightarrow${one
		boring→drilling(tapping)→	$C_y1P_lP_l$ $\_P_l\}\rightarrow${eight $C_y1P_lP_l\_P_l\}$
		drilling(tapping)	

Based on section 5, the similarities of geometry variation between two machining instances are calculated and shown in Tab. 3. In Tab. 3, the similarity of geometry variation between P_i ($0\leq i\leq 10$) and P_j ($0\leq j\leq 10$) is listed at the i^{th} row and the j^{th} column, and the values enclosed in red wireframes represent the similarities of geometry variation between two parts in the same type. As seen in Tab. 3, for the two parts belonging to the same type, their similarity of geometry variation caused by machining processes may be high. Meanwhile, for the parts belonging to different types, their similarity of geometry variation could be low.

Table 3. The similarity between any two geometry variation sequences

Sim	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
P_1	1.0000	0.6958	0.4961	0.4961	0.4961	0.3321	0.6006	0.1407	0.3570	0.6695
P_2		1.0000	0.6133	0.6133	0.6133	0.5904	0.7345	0.2286	0.4387	0.6681
P_3			1.0000	0.8498	0.8498	0.9313	0.6892	0.2167	0.6667	0.8498
P_4				1.0000	1.0000	0.7357	0.6518	0.1444	0.9027	0.5564
P_5					1.0000	0.6962	0.6978	0.0867	0.8061	0.5538
P_6						1.0000	0.5867	0.7202	0.2000	0.7237
P_7							1.0000	0.1487	0.5807	0.8129
P_8								1.0000	0.9548	1.0000
P_9									1.0000	0.6303
P_{10}										1.0000

7 Conclusions

To address the process reuse, a similarity measurement method for geometry variation sequence is proposed in this work. The similarity between processes is evaluated based on their geometry variation instead of the part's geometry themselves, so as to improve the reuse. The geometric construct of the part has also been taken into the consideration, which provides a new idea for process retrieval. The case studies show that proposed method has the potential to be applied directly in industry.

Declarations

Availability of data and materials

All data generated or analysed during this study are included in this manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2019JQ-896) and Research and Development Program of Shaanxi (Program No. 2019GY-091).

Authors' contributions

L C. was in charge of the whole trial and wrote the manuscript; L L. and W X. assisted with mathematical modeling. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

References

[1] Chang HC, Dong L, Liu FX, Lu WF (2000) Indexing and retrieval in machining process planning using case-based reasoning. Artif Intell Eng 14(1):1–13

[2] El-Mehalawi M, Miller RA (2003) A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching, and similarity assessment. Compu Aid Des 35(1):95–105

[3] Cuillière JC, François V, Souaissa K, Benamara A, Belhadjsalah H (2011) Automatic comparison and remeshing applied to CAD model modification. Compu Aid Des 43(12):1545–1560

[4] Zhang C, Chen T (2001) Indexing and retrieval of 3D models aided by active learning. ACM International Conference on Multimedia:615–616

[5] Hou S, Lou K, Ramani K (2005). Svm-based semantic clustering and retrieval of a 3d model database. Comput Aided Des Appl 2(1-4), 155–164

[6] Ohbuchi R, Kobayashi J. (2006) Unsupervised learning from a corpus for shape-based 3D model retrieval. ACM Sigmm International Workshop on Multimedia Information Retrieval:163–172

[7] Biundo S, Dengler D, Koehler J (1998) Deductive planning and plan reuse in a command language environment. European Conference on Artificial Intelligence, John Wiley & Sons:628–632

[8] Kambhampati S (1990) Mapping and retrieval during plan reuse: a validation structure based approach. Eighth National Conference on Artificial Intelligence, AAAI Press:170–175

[9] Liu SN, Zhang ZM, Tian XT (2007) A typical process route discovery method based on clustering analysis. Int J Adv Manuf Technol 35: 186–194

[10] Jiang Z, Jiang Y, Wang Y, Zhang H, Cao H, Tian G (2016) A hybrid approach of rough set and case-based reasoning to remanufacturing process planning. J Intell Manuf.

doi:10.1007/s10845-016-1231-0

[11] Alemanni M, Destefanis F, Vezzetti E (2011) Model-based definition design in the product lifecycle management scenario. Int J Adv Manuf Technol 52:1–14

[12] Huang R, Zhang S, Bai X (2014) Multi-level structuralized model-based definition model based on machining features for manufacturing reuse of mechanical parts. Int J Adv Manuf Technol 75:1035–1048

[13] Zhang X, Liang C, Li WY (2014) Automatic process intermediate model generation in process planning. Adv Mater Res 834-836:1436–1443.

[14] Zhang X, Liang C, Si T, Ding D (2013) Machining Feature Modeling and Process Intermediate Model Generation in Process Planning. ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference:V03BT03A010

[15] Zhu H, Zhou MC, Alkins R (2012) Group role assignment via a Kuhn–Munkres algorithm-based solution. IEEE Trans Sys Man Cyb A Sys Hum 42(3):739–750

[16] Yuan ZW, Zhang H (2012) Research on application of Kuhn-Munkres algorithm in emergency resources dispatch problem. International Conference on Fuzzy Systems and Knowledge Discovery, IEEE:2774–2777

[17] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local aligment search tool. J Mol Biol 215(3):403–410

Figures

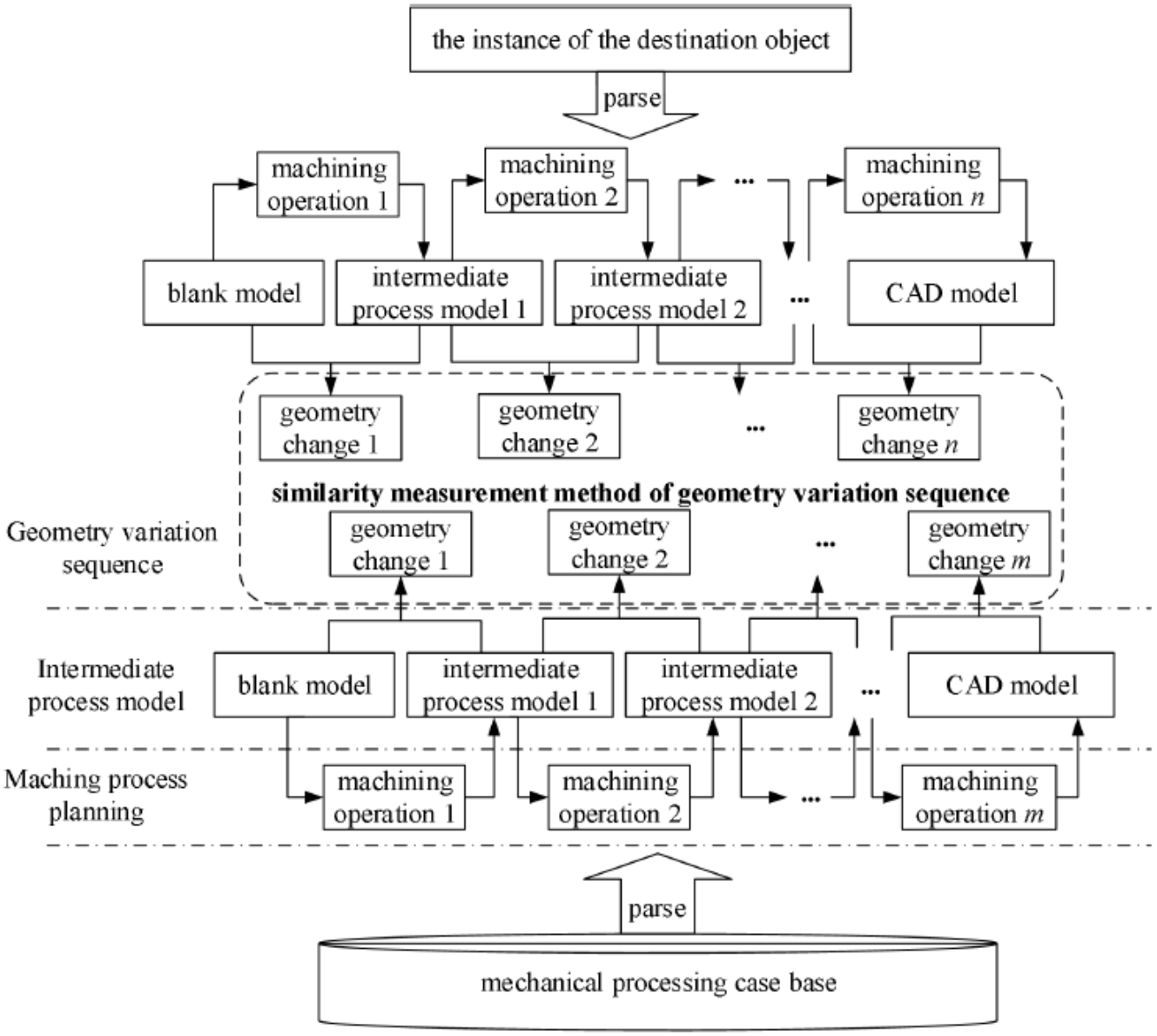

Figure 1

General framework of similarity measurement of geometry variation sequence

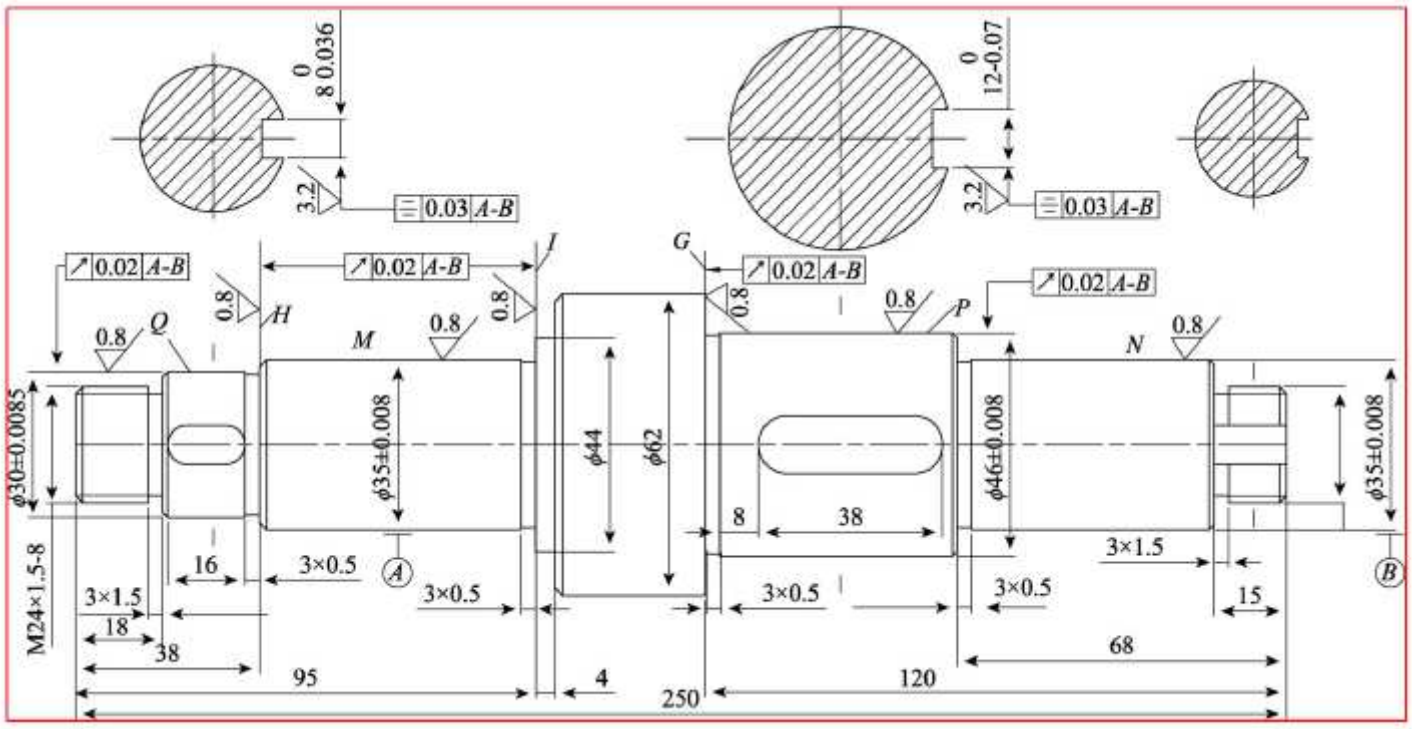

Figure 2

A transmission shaft part

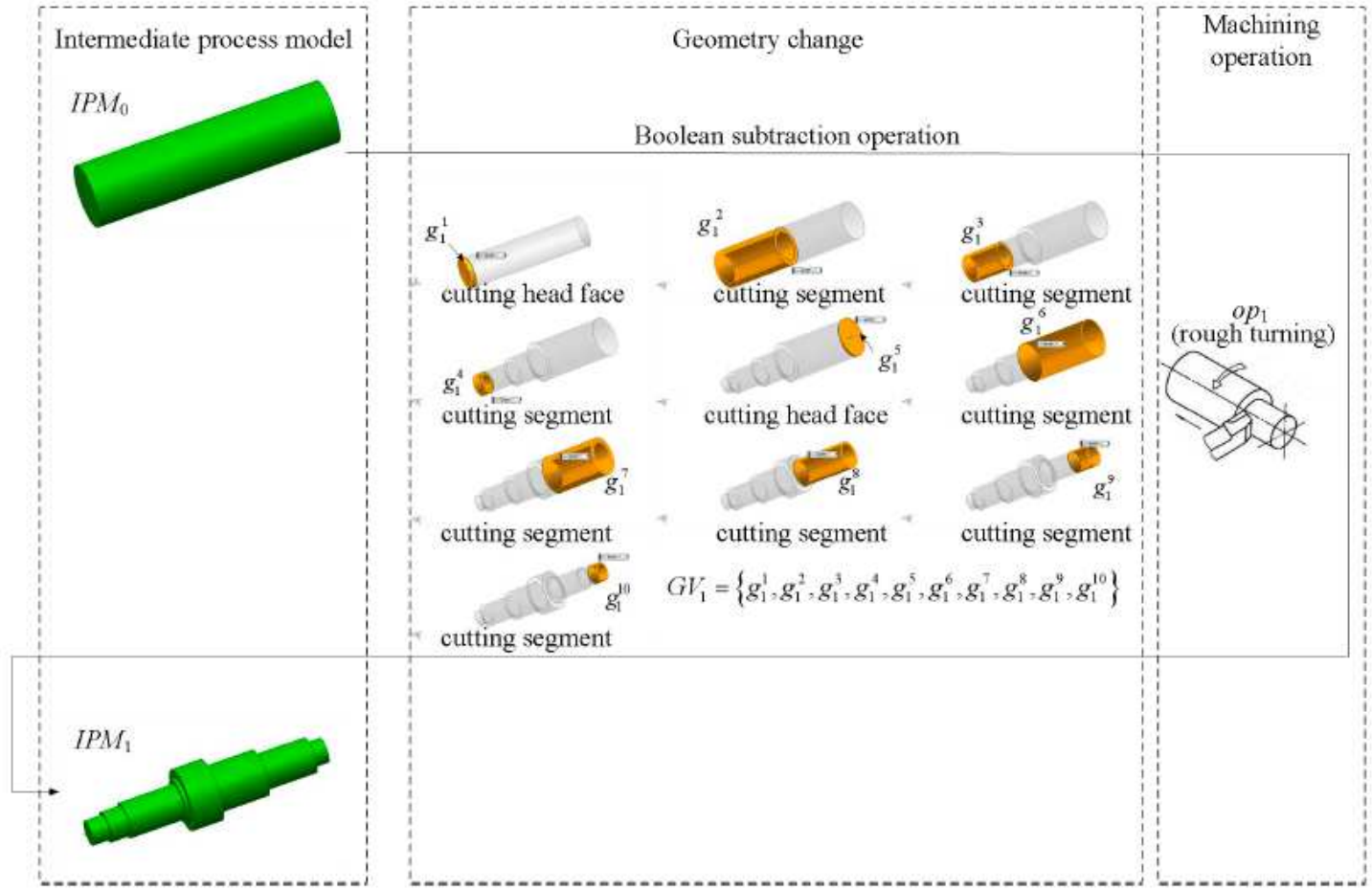

Figure 3

Extraction of geometry change

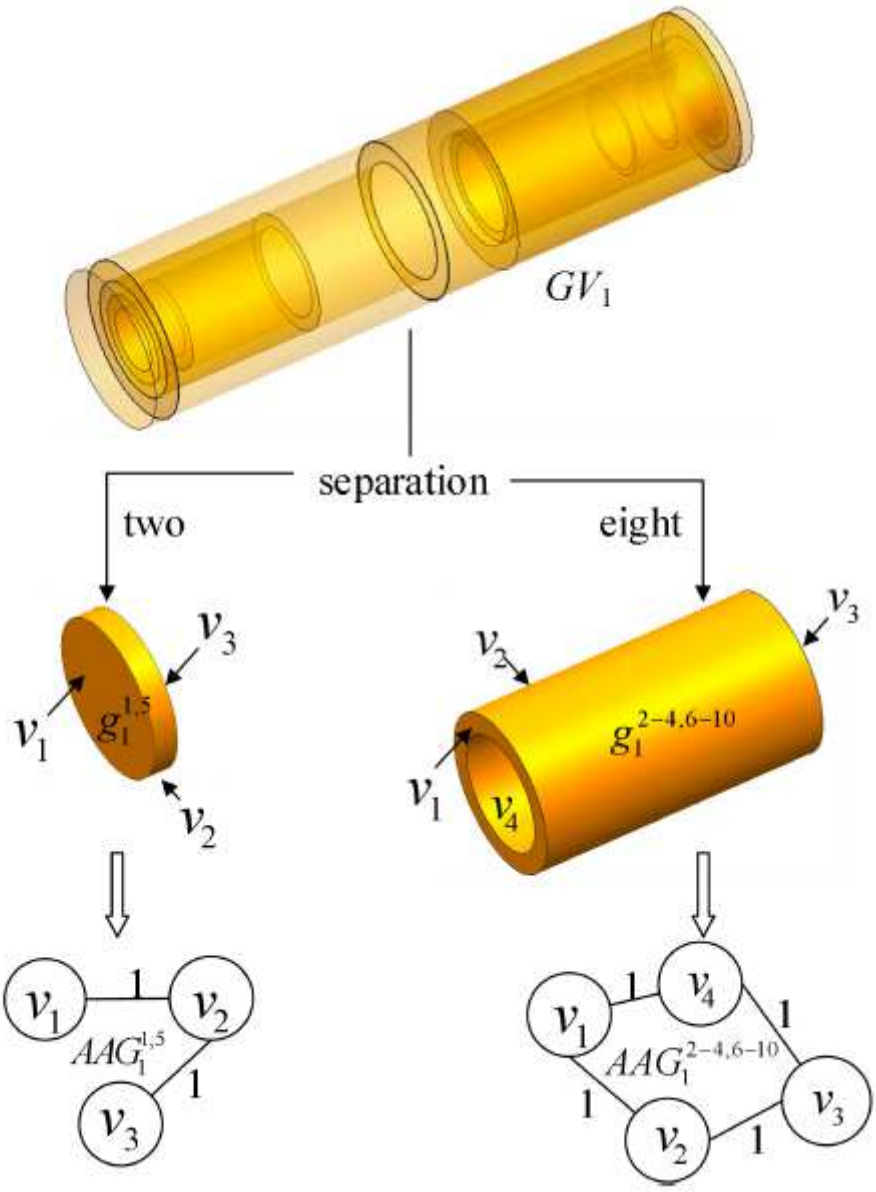

Figure 4

Graphical representation of geometry change

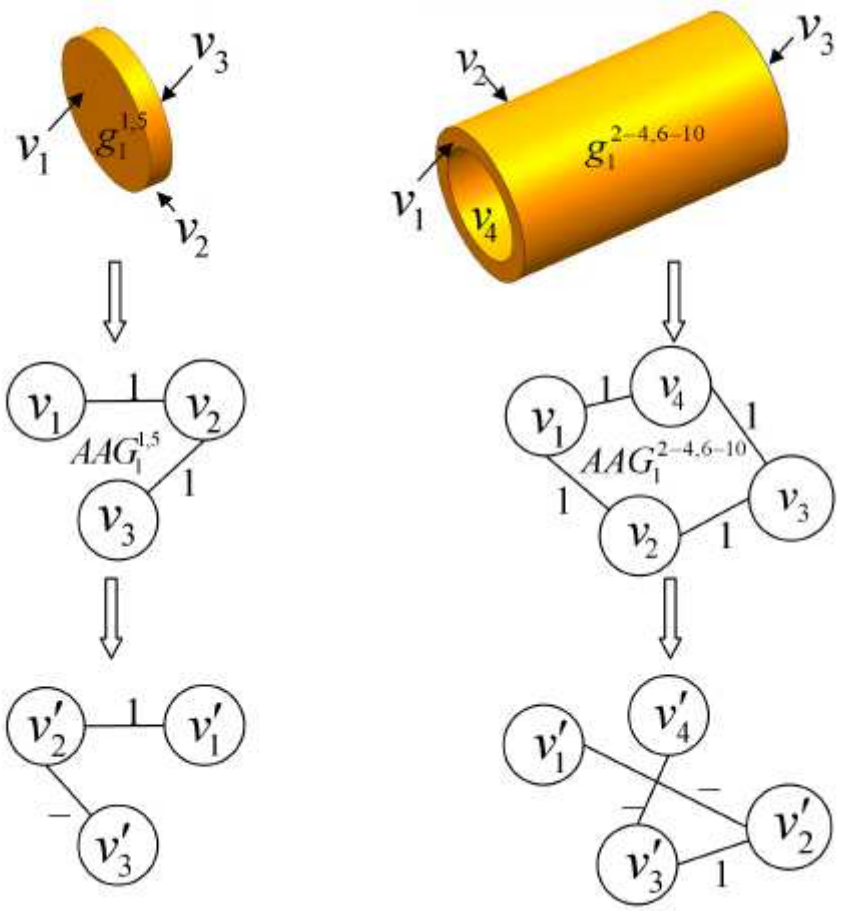

Figure 5

Attribute Adjacent Graph with sorted vertexes

	P_l	_	P_l	P_l	1	C_y	C_y	_	C_y
	0	1	2	3	4	5	6	7	8	9
C_y	1	1	2	3	4	5	5	6	7	8
1	2	2	2	3	4	4	5	6	7	8
P_l	3	2	3	2	3	4	5	6	7	8
P_l	4	3	3	3	2	3	4	5	6	7
_	5	4	3	4	3	3	4	5	5	6
P_l	6	5	4	3	4	4	4	5	6	6

Figure 6

Matching relation matrix Ld between cylinder and ring solid

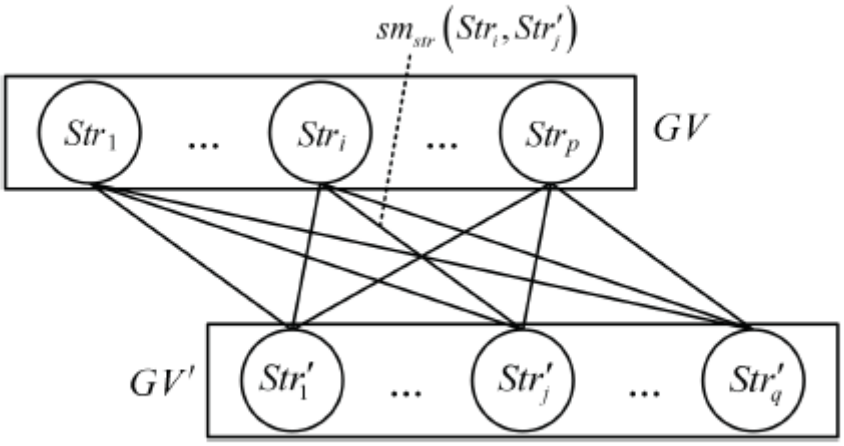

Figure 7

Construction of the bipartite graph

```
function t =
  gv_sim(gv1,gv2)
  len1=length(gv1);
  len2=length(gv2); T=[]; if
  len1<=len2 for i=1:len2
      c(i)=i;
  end
  M=nchoosek(c,len1);
  m=size(M,1); for u=1:m
  for i=1:len1 for j=1:len1
                  A(i,j)=(2/(len1+len2))*sm(gv1{i},gv2{M(u,j)});
              end
      end
      B=[];
      [B,T(u)]= Kuhn_Munkres(A,len1);
  end t=max(T); else for i=1:len1
      c(i)=i;
  end
  M=nchoosek(c,len2);
  m=size(M,1);
```

```
  for u=1:m for
      i=1:len2 for
      j=1:len2
              A(i,j)=(2/(len1+len2))*sm(gv1{M(u,i)},gv2{j});
          end
      end
      B=[];
      [B,T(u)]= Kuhn_Munkres(A,len2);
      end t=max(T);
  end
end
```

Figure 8

The pseudo-code of calculating similarity of geometry change

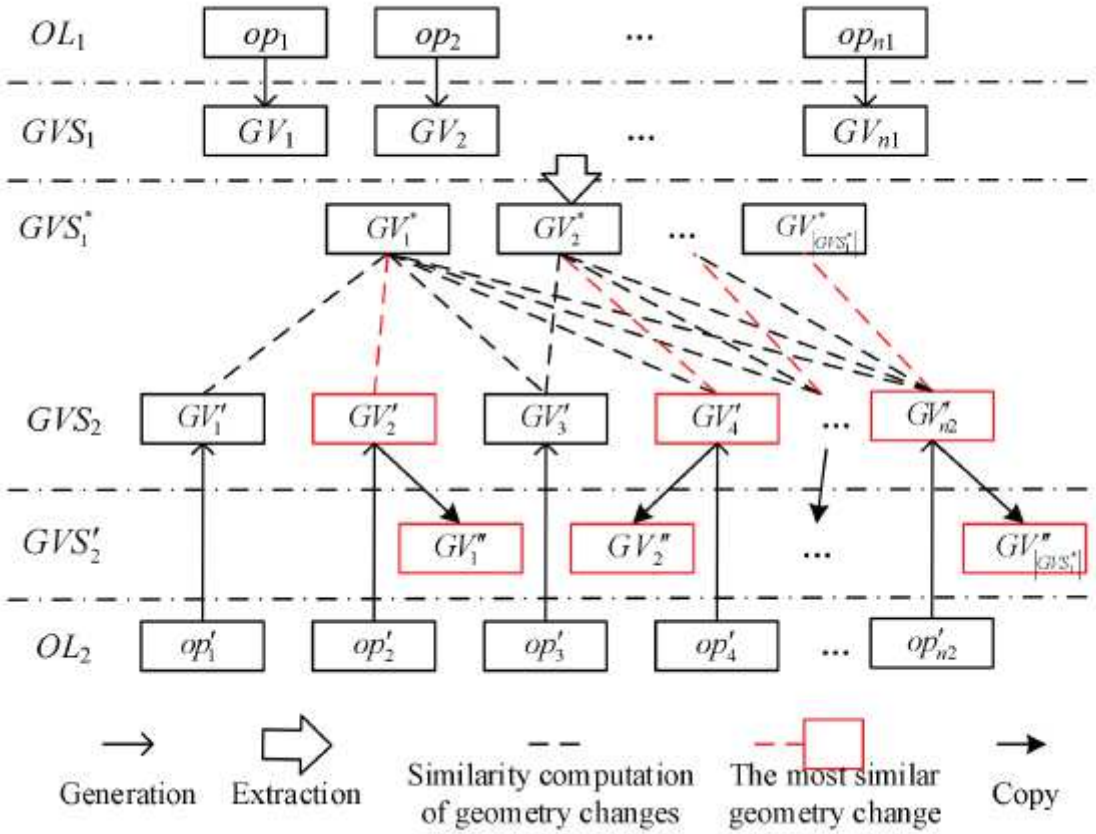

Figure 9

Acquisition process of the matching sequence

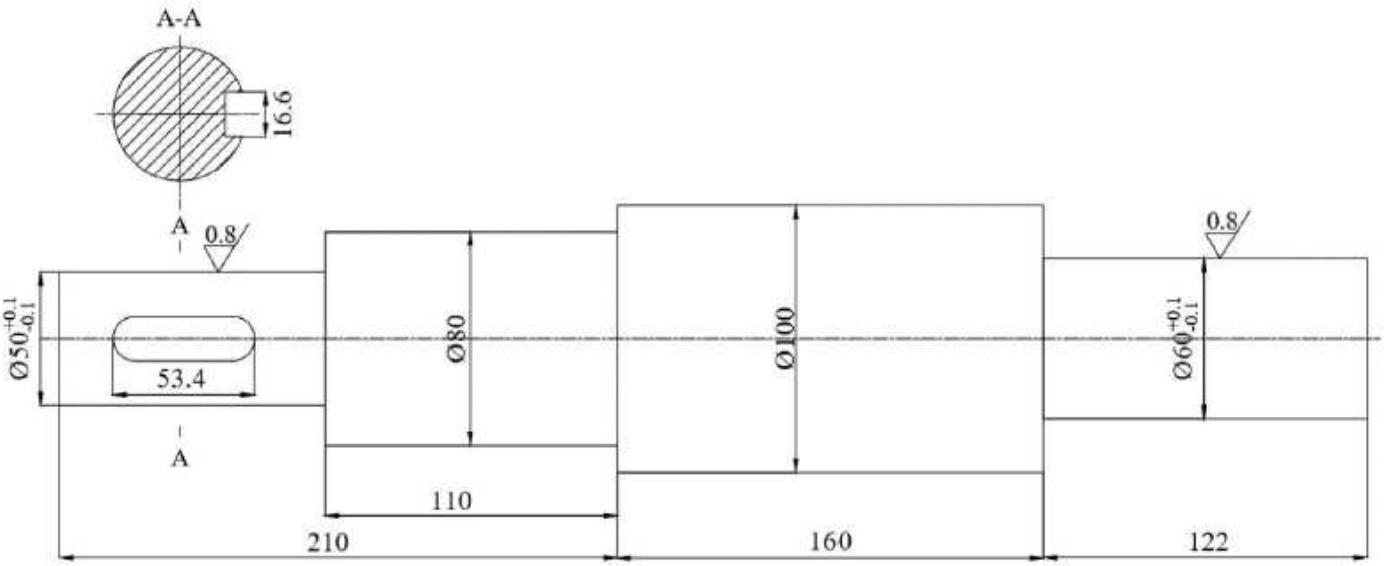

Figure 10

A stepped shaft

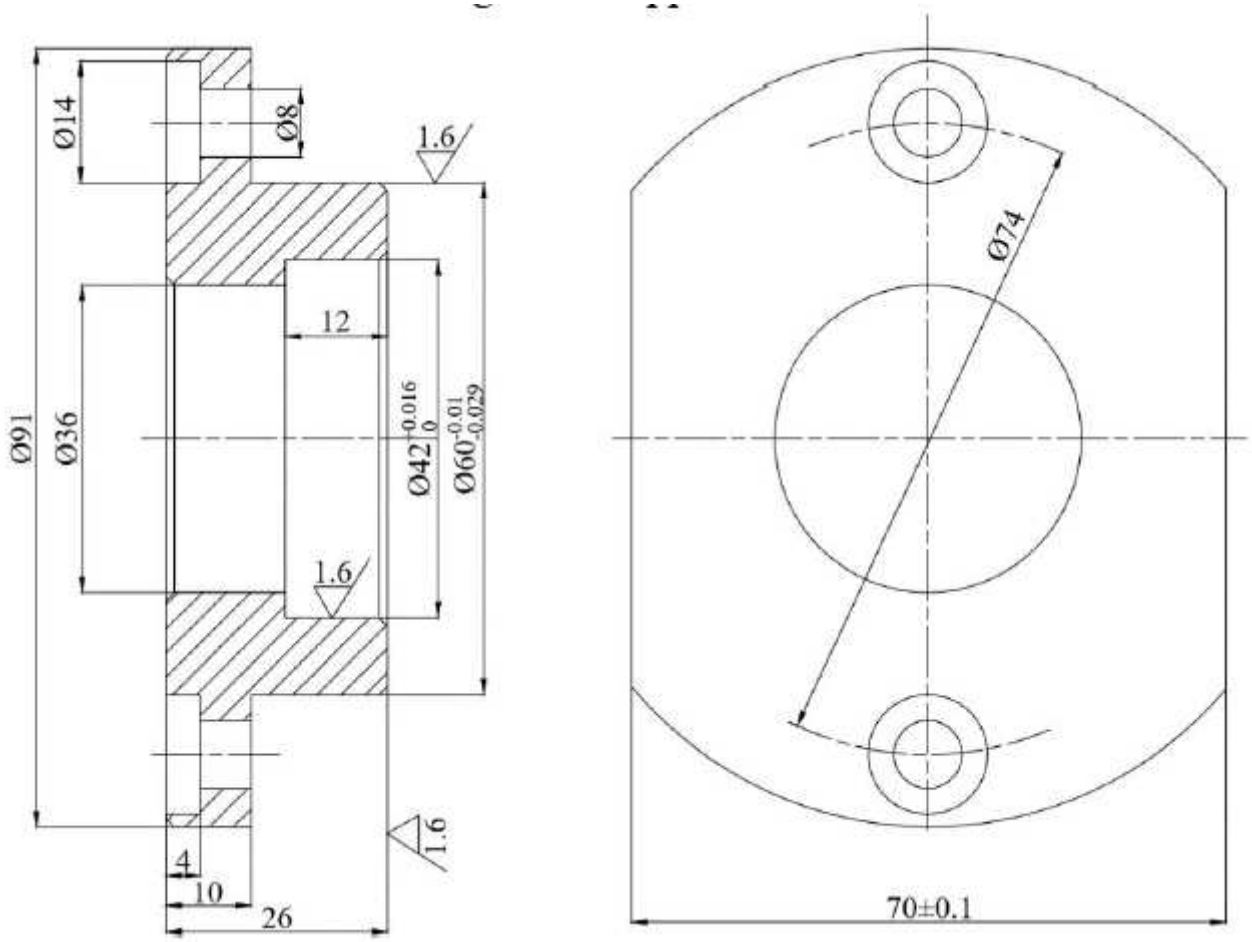

Figure 11

A cover part

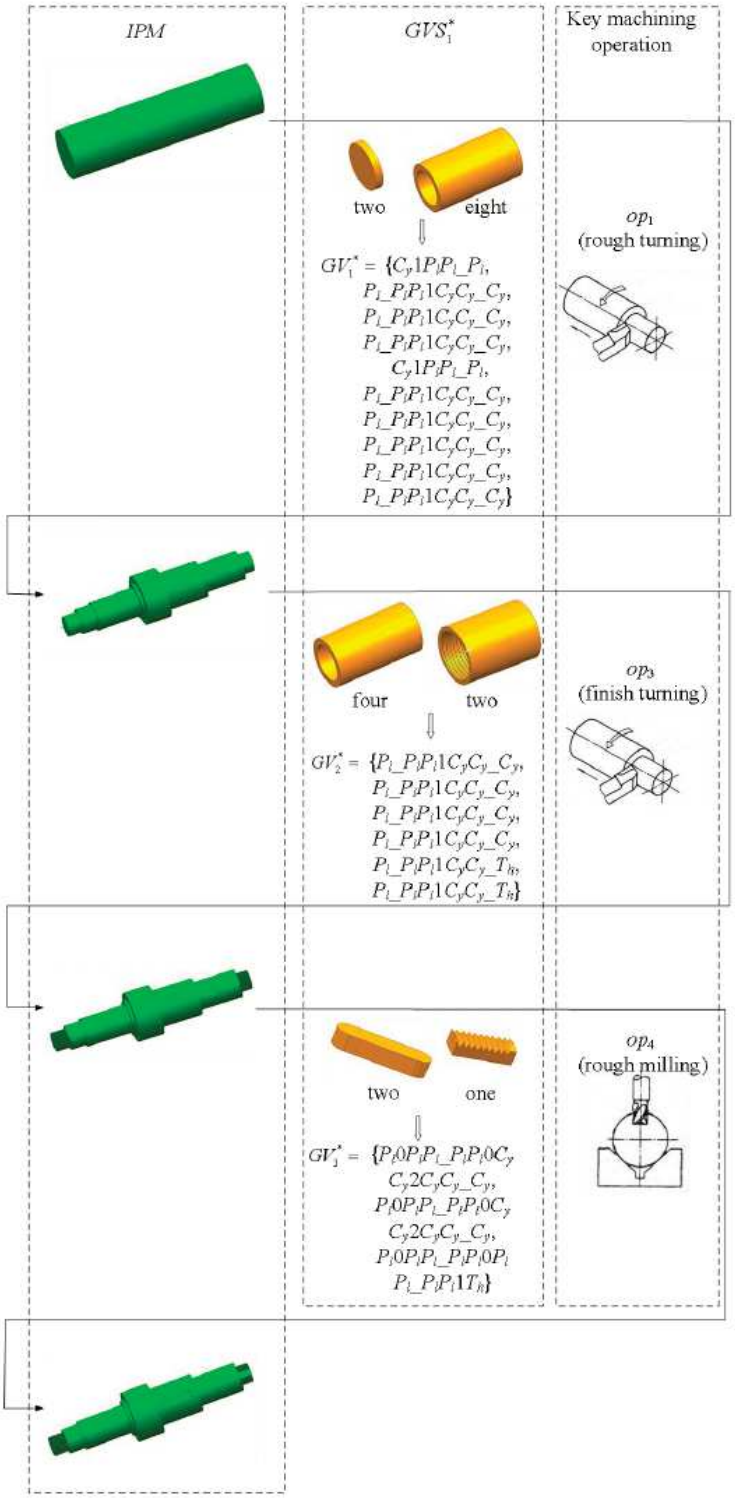

Figure 12

The key geometry variation sequence GVS1*

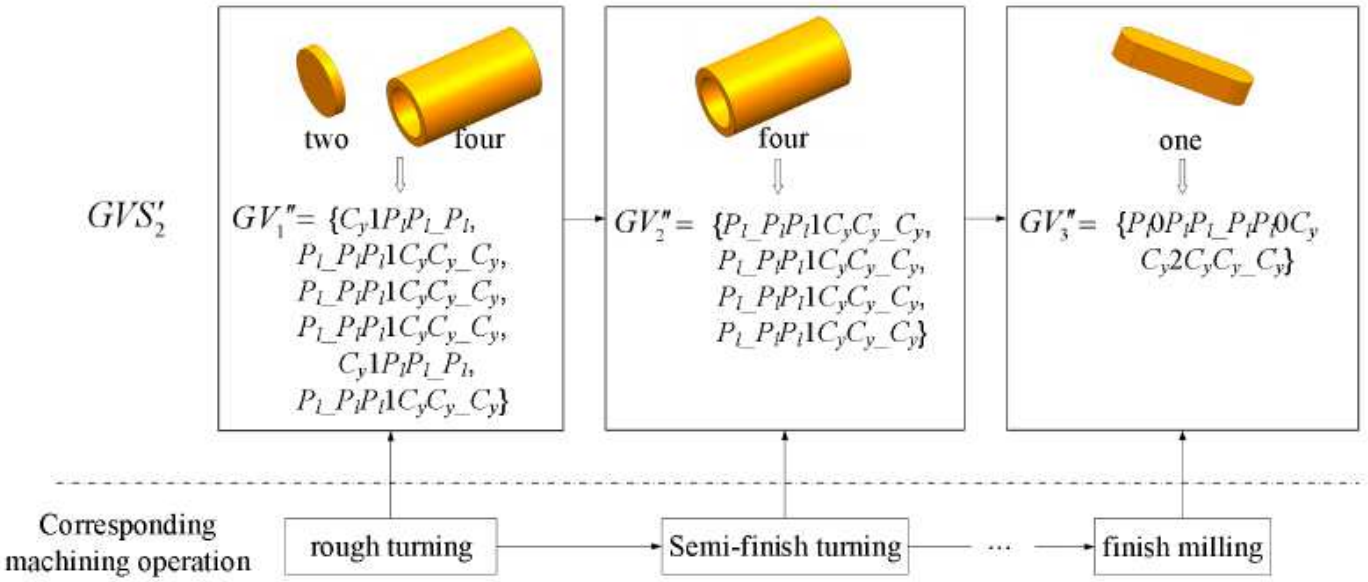

Figure 13

The matching sequence GVS2'

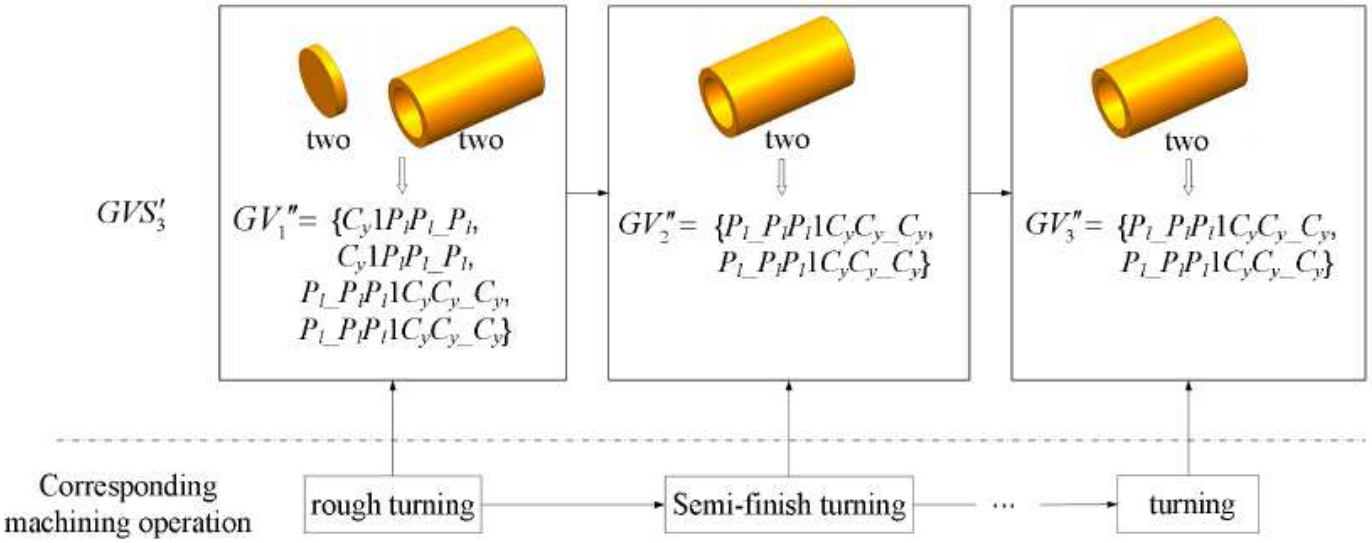

Figure 14

The matching sequence GVS3'