Production Systems and Information Engineering, Miskolc, Vol. 1. (2003), pp. 71-81

COMPUTERIZED SIMILARITY ANALYSIS OF PARTS FOR SUPPORTING GROUP TECHNOLOGY

TIBOR TÓTH*
Department of Information Engineering, University of Miskolc
H-3515. MISKOLC- Hungary
toth@iit.uni-miskolc.hu

MIHÁLY MOLNÁR*
Department of Information Engineering, University of Miskolc
H-3515. MISKOLC- Hungary
molnar@iit.uni-miskolc.hu

[Received March and accepted October 2002]

* Production Information Engineering Research Team (PIERT) of the Hungarian Academy of Sciences; Department of Information Engineering of the University of Miskolc

Abstract. Group Technology (GT) is a comprehensive organizing principle of part manufacturing which can be utilized both in design and technology processes of parts as well as in the course of real manufacturing processes, too. The base of GT is *similarity* of the most important macro-geometrical and technological features of parts. The paper deals with grouping the parts according to similarity which is the first major activity area indispensable for initiating GT. It shows an algorithm suitable for solving two basic tasks: (1) To form groups from a given part set automatically such a way that within each group (class) the deviation between the numerical values allocated to two optional parts on the base of the same similarity parameters should be less than a given value; (2) To insert a new part into that group (class) which is the most adequate to it on the base of its properties taking into consideration a previously ordered part set and to create a new group (class) if inserting is not successful, respectively.The paper also outlines the fuzzy method constituting the mathematical base of the algorithm and gives a brief summary about the computer program realizing the algorithm in the present phase of development.

Keywords: Group Technology (GT), part classification, similarity matrix, fuzzy method

1. Introduction

Industrial, commercial and service changes in the last quarter of the twentieth century stimulated enterprises to increase the flexiblility of production. The adequate reply to this challenge was the unification and systems approach based integration of the means of production engineering and information technology. This kind of integration results in development of Computer Integrated Manufacturing (CIM) systems. Group Technology, as a comprehensive organizing principle and methodological tool of part manufacturing, plays a significant role in CIM systems.

Group Technology (GT), according to the definition of Ryerson Polytechnic (Toronto, Canada), is „... the philosophy of recognizing the similarities between entities (problems, parts, machines, designs, processes etc) and utilizing this knowledge." [1]. Another useful definition: „Group Technology is the idea of studying a great population of apparently different items and then dividing them into groups of items having the same or similar characteristics." [2].

This paper only deals with dividing machined parts into groups of parts on the base of similarity characteristics. In production process, starting from production planning and going through design phase, process planning as well as the different phases of real manufacturing process (e.g.: prefabrication, part manufacturing, assembly, storage) to the final quality control, the activities are oriented to more or less similar individual parts [2], [3]. These parts can be ordered in a systematical way and the ordered sets, with finite number of elements for each, make it possible to apply procedures suitable for rationalization and optimization.

The group is a set of parts similar to each other to a certain extent and in some kind of sense and the individuals, because of their similar characteristics, make common design, planning, recording, storing, moving, machining etc. activities possible.

The method of Group Technology could not be very much used before seventies of the last century. The reason was the lack of the computer background suitable for supporting computational tasks. The computer-based methods spreading since that time, however, have improved efficiency and usefulness of the GT method to a considerable extent [2], [4]:

- creating the complex part models has accelerated significantly;
- decomposition of the part family trees can be executed in an efficient and fast way;
- velocity of the searching procedures based on similarity relations has increased to a great extent;
- the principles of grouping, building and decomposing can be extended to the manufacturing process as a whole;
- data bases and processing procedures can be unified, therefore compatibility of the interfaces can be ensured;
- GT applications have been supported by effectual software modules of general purpose and there has been a possibility to co-operate with them, respectively (e.g.: CAD systems, database handling systems).

Spreading of GT has been promoted by the essential changes in environmental conditions. They are as follows [2], [4]:

- product change has accelerated, demand for the modified, variable products has been increasing, i.e. demand for parts design has been increasing;
- flexible manufacturing requires frequented and fast change-over of series, therefore demand for the technology process planning of parts has also beenincreasing;
- the shortening through-put times require a technical preparation of higher and higher quality level, thus, in accordance with this fact, the demand for design of manufacturing tools, fixtures and jigs has been increasing, too;
- economical manufacturing needs an analytical supervising all the circumstances in connection with the manufacturing process itself to reveal reserves. To carry out this task, GT applications can provide a significant aid;
- GT-applications have been supported by computational technology, especially spreading of the local area networks and workstations. This support is more and more effective and is of smaller and smaller specific cost.

The advantages of GT, taking into consideration the aforementioned aspects, are as follows [2], [4]:

- GT promotes the integration of design, planning and manufacturing;
- GT guarantees a higher level of typifying, standardizing and harmonizing the cost calculation and loading procedures in comparison with the previous situation;
- GT decreases the costs of set-up, tooling and fixturing;
- GT makes it possible to decrease the through-put times and costs of parts design and process planning;
- GT contributes to standardization and decreases redundant redesigning the elements and parts with the same functions;
- GT helps to solve very complicated and hard-to-survey problems both in design and planning as well as manufacturing, on the base of systems approach.

2. Part classification by means of fuzzy methods

The algorithm and computer program to be shown in the paper should be suitable for solving two tasks connecting with each other. As an initial siutation let us consider the following:

It is given parts n $(n \in N)$ having features p $(p \in N)$ previously defined. Let $X_1, X_2, \ldots, X_n$ the identifiers of the parts and let us consider that the features of each part are expressed by real numbers p. So we can represent the part $X_i (1 \leq i \leq n)$ by means of a vector $(x_{i,1}, x_{i,2}, \ldots, x_{i,p}) \in R^P$ the components of which are the values of the features $1,2,..,p$ concerning the part X_i. In the course of classification of the parts n the following tasks are to be executed:

Task1 (grouping task): The parts, i.e. the vectors should be grouped such a way that in a group (class) the deviation between the same feature values of two optional parts be under a given limit, i.e. it is to be determined how many groups (classes) can be formed and which parts are in the detached groups (classes).

Task2 (inserting task): Suppose that a new member X_{n+1} will be added to the parts set ordered into the groups (classes) $X_1, X_2, \ldots, X_n$. The purpose is to insert the part X_{n+1} in the group (class) adequate to the features of it to the greatest extent, or, if this is not possible, to put it into a separated group (class).

The authors suggest a fuzzy method for ordering the parts into groups (classes). Conventional and fuzzy sets are differ from each other in that in the case of conventional sets a certain element either belongs to the set or not, however, in the case of fuzzy sets the measure of belonging to is expressed by means of a real number between 0 and 1. Let U be a set well-defined in conventional sense. The fuzzy set A is formed from the set U by means of a function $\mu_A : U \rightarrow [0,1]$ named *membership function*. (Accordingly, μ_A allocates to the elements of U real numbers in the interval [0,1]).

Let $\alpha \in [0,1]$ a fixed real number. In the case of a fuzzy set $A=(U,\mu_A)$, the totality of those elements $x \in U$ for which $\mu_A(x)=\alpha$, is named a *level* of the fuzzy set. The name of the set $A_\alpha = \{x \in U | \mu_A(x) \geq \alpha\}$ is a section α of the fuzzy set in question. We introduce two methods. The ***Method1*** examines how many groups (classes) can be formed on the base of the ***Task1***. For each part $X_i (1 \leq i \leq n)$ and feature $j (1 \leq j \leq p)$ we define a real number $\mu_{i,j} \in [0,1]$ to express that the part X_i has the feature j to what extent, i.e. we allocate to the parts set

$\{X_1, X_2, \ldots, X_n\}$ known vectors $(\mu_{1,1}, \ldots, \mu_{1,p}), \ldots, (\mu_{n,1}, \ldots, \mu_{n,p})$ where the components of the vectors can only receive real values between *0* and *1*.

We can interprete the „distance" of two optional parts X_i and X_j according to the next formula [4], [5]:

$$S_{i,j} = \frac{\sum_{k=1}^{p} \min\left(\mu_{i,k}, \mu_{j,k}\right)}{\frac{1}{2} \sum_{k=1}^{p} \left(\mu_{i,k} + \mu_{j,k}\right)} \tag{1}$$

From these distances we can form the *similarity matrix* of the parts in the following manner:

$$\mathbf{S} = \begin{bmatrix} 1 & S_{1,2} & S_{1,n} \\ S_{2,1} & 1 & S_{2,n} \\ & & \\ S_{n,1} & S_{n,2} & 1 \end{bmatrix}. \tag{2}$$

The similarity matrix **S** is symmetric to the principal diagonal and the value of all the elements in the principal diagonal is *1*.

To carry out ordering the parts into groups (classes) on the base of the similarity matrix **S** we have to introduce a new mathematical operation (°). If **R** and **T** are fuzzy relations interpreted in the same set of elements *n* then the matrix of the *product relation* **R°T** can be obtained from the matrices **R** and **T** by means of multiplying, where this mathematical operation is not the conventional matrix multiplying but an operation defined as follows:

$$\underset{\mathbf{R}}{\begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,n} \\ r_{2,1} & r_{2,2} & r_{2,n} \\ & & \\ r_{n,1} & r_{n,2} & r_{n,n} \end{bmatrix}} \circ \underset{\mathbf{T}}{\begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,n} \\ t_{2,1} & t_{2,2} & t_{2,n} \\ & & \\ t_{n,1} & t_{n,2} & t_{n,n} \end{bmatrix}} = \underset{\mathbf{E}}{\begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,n} \\ e_{2,1} & e_{2,2} & e_{2,n} \\ & & \\ e_{n,1} & e_{n,2} & e_{n,n} \end{bmatrix}} \tag{3}$$

where we can get to an element $e_{i,j}$ of the result matrix in the following way:

$$e_{i,j} = \max[\min(r_{i,k}, t_{k,j})]. \quad (k=1, \ldots, n) \tag{4}$$

The matrix $\mathbf{R}_{n-1}$ is derived from the similarity matrix **S** as follows:

$$\mathbf{R}_{n-1} = \underbrace{\mathbf{S} \circ \cdots \circ \mathbf{S}}_{(n-1)-times} \tag{5}$$

where *n* is the number of the parts.

The elements of the matrix $\mathbf{R}_{n-1}$ express similarity between the parts. If $R_{i,j}=0$ then part i and part j are completely different and else if $R_{i,j}=1$ then part i and part j are completely alike.

After having constructed the matrix $\mathbf{R}_{n-1}$, we can give different treshold values for representing the similarity of the parts (e.g. the parts will be ranged in the same class if they are similar to each other in 80 % at least, i.e. $\alpha=0.8$). In this case the elements can be ranged into classes through an immediate reading-out of the matrix $\mathbf{R}_{n-1}$ (***Method1***).

Example: Let the parts set $X=\{X_1,X_2,X_3,X_4,X_5,X_6\}$ is given. The matrix $\mathbf{R}_{n-1}$ derived from the similarity matrix **S** is as follows:

$$\mathbf{R}_{n-1}=\begin{bmatrix} 1 & 0.2 & 1 & 0.6 & 0.2 & 0.6 \\ 0.2 & 1 & 0.2 & 0.2 & 0.8 & 0.2 \\ 1 & 0.2 & 1 & 0.6 & 0.2 & 0.6 \\ 0.6 & 0.2 & 0.6 & 1 & 0.2 & 0.8 \\ 0.2 & 0.8 & 0.2 & 0.2 & 1 & 0.2 \\ 0.6 & 0.2 & 0.6 & 0.8 & 0.2 & 1 \end{bmatrix}. \qquad (6)$$

Taking different values for $\alpha\in[0,1]$ we have different partitions of the set X on the base of the matrix $\mathbf{R}_{n-1}$. If $\alpha=1$ then the number of the classes will be the highest one, because the completely identical elements can only be ranged into the same class. In the case of example (6) the set X will consist of 5 partitions. They are as follows: $\{X_1,X_3\};\{X_2\};\{X_4\};\{X_5\};\{X_6\}$. If $\alpha=0$ then the number of the classes will be 1 exactly, since all the elements will be ranged into one class. It is easy to see that in the other cases the number of the classes will be equal to the maximum number of the classes or less than the maximum number. E.g. in case of $\alpha=0.8$ the elements of the set X can be arranged in 3 classes. They are as follows: $\{X_1,X_3\};\{X_4,X_6\};\{X_2,X_5\}$.

Solving the ***Task1*** does not make it possible to order the elements in optional number of classes therefore we have to modify the ***Task1*** such a way that we give not the similarity trashold value (α) but an integer number $c\in N$ which indicates that the elements $X_1,X_2,\dots,X_n$ are to be grouped into how many classes (***Task1****).

The ***Method2*** which is suitable for solving the ***Task1**** is based on the concept of fuzzy-classification and an iterative procedure. As is known that in the case of conventional (non-fuzzy) classification an element either belongs to a given class or not, there exists no other case. However, the fundamental principle of fuzzy classification is that an element belongs to a given class in a certain percentage which can be expressed by means of a real number within the interval [0,1]. In accordance with this, we define the number $u_{i,j}$ which means that the element X_j belongs to the i-th class to what extent. By this means we have the matrix **T** as follows:

$$\mathbf{T} = \begin{array}{cc} & \begin{array}{cccc} X_1 & X_2 & \quad & X_n \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \dots \\ c \end{array} & \begin{bmatrix} u_{1,1} & u_{1,2} & \quad & u_{1,n} \\ u_{2,1} & u_{2,2} & \quad & u_{2,n} \\ & & & \\ u_{c,1} & u_{c,2} & \quad & u_{c,n} \end{bmatrix} \end{array}. \tag{7}$$

For the matrix **T** the following constraints are valid:

$$\begin{aligned} &0 \le u_{i,j} \le 1, \quad i = 1,2,\dots,c \quad j = 1,2,\dots,n \\ &\sum_{i=1}^{c} u_{i,j} = 1, \quad j = 1,2,\dots,n \\ &\sum_{j=1}^{n} u_{i,j} > 0, \quad i = 1,2,\dots,c \end{aligned} \tag{8}$$

For solving the ***Task1**** we arrange the known vectors $(\mu_{1,1}, \mu_{1,2} \dots, \mu_{1,p}), \dots (\mu_{n,1}, \mu_{n,2}, \dots, \mu_{n,p})$ in an nxp type matrix **M**:

$$\mathbf{M} = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \quad & \mu_{1,p} \\ \mu_{2,1} & \mu_{2,2} & \quad & \mu_{2,p} \\ & & & \\ \mu_{n,1} & \mu_{n,2} & \quad & \mu_{n,p} \end{bmatrix} \tag{9}$$

Let us denote the classes of the decomposition to be obtained by $C_1, C_2, \dots, C_c$. If we would know the matrix **T** of cxn size then we could determine the fictitious feature values $1,2,\dots,p$ correspond to the classes $C_1, C_2, \dots, C_c$ in the following manner:

$$\begin{aligned} &v_{i,k} = \frac{\sum_{j=1}^{n} u_{i,j}^{m} \ \mu_{j,k}}{\sum_{j=1}^{n} u_{i,j}^{m}} \\ &i = 1,\dots,c \\ &k = 1,\dots,p \end{aligned} \tag{10}$$

where $m>1$ is a rational number previously fixed.

On the base of formula (10) we would get a vector $V_i=(v_{i,1},\ v_{i,2}, \quad v_{i,p})$ that would express to what extent class C_i has the features $1,2,\dots,p$.

Hence, the steps of the procedure needed can be summarized as follows [4], [5]:

(1) Let us define a matrix

$$\mathbf{T}_0 = \left[\mu^0_{i,j}\right] \tag{11}$$
$$i = 1,\ldots,c \quad \text{and} \quad j = 1,2,\ldots,n$$

such a way that the elements of which as optional values should meet the constraints (8).

(2) On the base of formula (10) we determine the values $v_{i,k}$ for every i=1,2,…,c and k=1,2,…,p, i.e. we determine the vectors V_i.

(3) By using formula (12), see below, we determine the new value $u'_{i,j}$ of the marix **T** and fill up matrix $\mathbf{T}_0$ with the $u_{i,j}$ values before the iterative step. In the course of iterative process, from the beginning to end, we need two matrices **T** and $\mathbf{T}_0$ such a way that $\mathbf{T}_0$ includes always the values of the previous step.

$$u'_{i,j} = \frac{1}{\sum_{l=1}^{c}\left[\frac{\sum_{k=1}^{p}\left(\mu_{j,k} - v_{i,k}\right)^2}{\sum_{k=1}^{p}\left(\mu_{j,k} - v_{l,k}\right)^2}\right]^{\frac{1}{m-1}}}. \tag{12}$$

(4) If $K = \{\max | u_{i,j} - u^0_{i,j} |,\ i = 1,\ldots,c\ ,j = 1,\ldots,n\} \le \varepsilon$, where ε is an error-limit given by us, we stop else return to the step (2). The process will be repeated until $K \le \varepsilon$, i.e. until determination of optimum $u_{i,j}$ ($i=1,\ldots,c$, $j=1,\ldots,n$) with ε error tolerance.

(5) Element X_j will be ranged into class C_i if $u_{i,j}$ is the maximum of $u_{1,j}, u_{2,j}, \ldots, u_{c,j}$ values, i.e.

$$u_{i,j} = \max\{u_{k,j} \mid 1 \le k \le c\} \tag{13}$$

After these we determine the last $V_1, V_2, \ldots, V_c$ vectors featuring the classes C_1, $C_2, \ldots, C_c$ on the base of formula (10).

The ***Method1*** does not give an acceptable result if the number of parts n is large, at the same time, it is more flexible than ***Method2.*** Therefore it is expedient to combine the two methods. Firstly, on the base of ***Method1***, an adequate number of classes will be determined, i.e. the number of classes c will be calculated by means of matrix **S** for a certain similarity threshold value α previously given. After this ***Method2*** will be applied which results in much more correct fuzzy partition.

To solve ***Task2***, i.e. to insert the part X_{n+1} into the adequate group, first the parts $X_1, X_2, \ldots, X_n$ should be arranged into groups (classes) according to ***Method2***.

Let us allocate the vector $(\mu_{n+1,1}, \ldots, \mu_{n+1,p})$ to the part X_{n+1} and the vectors

$$V_1 = (v_{1,1}, v_{1,2}, \ldots, v_{1,p})$$
$$V_2 = (v_{2,1}, v_{2,2}, \ldots, v_{2,p})$$

$$V_c = (v_{c,1}, v_{c,2}, \ldots, v_{c,p})$$

to the classes $C_1, C_2, \ldots, C_c$ according to the formula (10) which expresses that the classes $C_1, C_2, \ldots, C_c$ have the features $1, 2, \ldots, p$ to what extent.

Let us denote a threshold value by t which shows above what distance the part in question can be allocated to a given class. This threshold value is constant for every classes. If

$$\left(X_{n+1}, V_i\right) = \max\{(X_{n+1}, V_k) \mid 1 \le k \le c \text{ and } (X_{n+1}, V_i) \ge t\} \tag{15}$$

then X_{n+1} can be inserted into the class C_i featured by the vector V_i because this class is the nearest one taking into consideration all the features $1, 2, \ldots, p$.

If X_{n+1} can be inserted into neither of classes it can be arranged in a new class or the equivalence analysis can be repeated for the part set of $(n+1)$ elements $\{X_1, X_2, \ldots, X_n, X_{n+1}\}$ because it is possible that the „distances" of X_{n+1} from the other $X_1, X_2, \ldots, X_n$ elements are not the same. So, in the course of a new classification, it would be possible to insert it into a certain class.

1. A computer program for part classification and inserting a new part

The program suitable for solving ***Task1*** and ***Task2*** is not an idependent one but it is a part of a larger application. Therefore the program has not yet own user interface at present but operation of the algorithm and computing can be traced by means of a „test-option" built-in the program and the input data as well as the results are sstored in text files readable in an easy way.

Starting the program can be carried-out in four kinds of way:

„fuzzy_gt" (arrangement of n parts into groups)

„fuzzy_gt-t" (arrangement of n parts into groups and writing the partial results on the screen)

„fuzzy_gt new" (inserting the $(n+1)$-th part)

„fuzzy_gt new-t" (inserting the $(n+1)$-th part and writing the partial results on the screen).

Solving the Task1 and Task1*

Solving the tasks is carried-out on the base of the process chart seen in Fig.1. The input data have to be given in the file *fuzzy_in.-dat* as follows:

Number of rows:		*Data:*
1	N	the number of the parts to be arranged into groups
2	p	the number of features for a part
3	α	similarity threshold value
4	c	the number of groups (classes)
5	0	
6	0	
7	0	
8	0	
9	0	
10	0	
11	$\mu_{1,1}$	The 1st feature of the 1st part ($\mu_{1,1} \in [0,1]$).
	$\mu_{1,2}$	The 2nd feature of the 1st part ($\mu_{1,2} \in [0,1]$)

	$\mu_{2,1}$	The 1st feature of the 2nd part ($\mu_{2,1} \in [0,1]$)
	$\mu_{2,2}$	The 2nd feature of the 2nd part ($\mu_{2,2} \in [0,1]$)
	$\mu_{n,1}$	The 1st feature of the n-th part ($\mu_{n,1} \in [0,1]$)
	$\mu_{n,p}$	The p-th feature of the n-th part ($\mu_{n,p} \in [0,1]$).

If $c=0$ then the program determines the number of groups (classes) c on the base of the similarity matrix **S** and the threshold value α .

The result of the analysis can be found in the file *fuzzy-ou.dat* the structure of which is as follows:

Numbers of rows:		*Data:*
1	c	the number of groups (classes)
2	$X_{1,1}$	The 1st part is in the 1st group (0 or 1)
	$X_{1,2}$	The 2nd part is in the 1st group (0 or 1)
	$X_{1,n}$	The n-th part is in the 1st group (0 or 1)
	$X_{2,1}$	The 1st part is in the 2nd group (0 or 1)
	$X_{2,n}$	The n-th part is in the 2nd group (0 or 1)
	$X_{c,1}$	The 1st part is in the c-th group (0 or 1)
	$X_{c,n}$	The n-th part is in the c-th group (0 or 1)

Solving the Task2

To solve ***Task2*** we have to know the data of ($n+1$)-th part. Giving the data of n part is the same as it was previously. Data of the ($n+1$)-th part have to be given in the file *fuzzy_i2.dat* as follows:

Number of rows:		*Data:*
1	$\mu_{n+1,1}$	The 1st feature of the $(n+1)$-th part ($\mu_{n+1,1} \in [0,1]$)
2	$\mu_{n+1,2}$	The 2nd feature of the $(n+1)$-th part ($\mu_{n+1,2} \in [0,1]$)
	$\mu_{n+1,p}$	The p-th feature of the $(n+1)$-th part ($\mu_{n+1,p} \in [0,1]$).

Computing is carried-out on the base of the flow chart seen in Fig.1 and the result will be stored in the file *fuzzy_02.dat* which has the following structure:

Number of rows:		*Data:*
1	$X_{n+1,1}$	The $(n+1)$-th part is in the 1st group (0 or 1)
2	$X_{n+1,2}$	The $(n+1)$-th part is in the 2nd group (0 or 1)
	$X_{n+1,c}$	The$(n+1)$-th part is in the group c (0 or 1)
	$X_{n+1,c+1}$	The $(n+1)$-th part has to be arranged into a new group ($c+1$) (0 or 1).

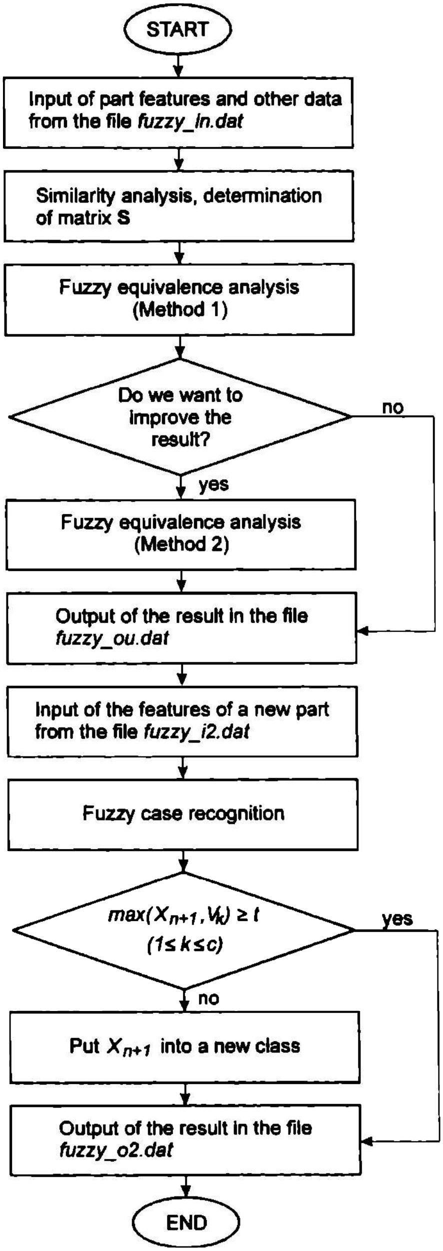

Fig.1.: The flow chart of the computer program suitable for similarity analyis

2. Acknowledgements

The work on which this paper is based has been supported by the OTKA (Hungarian Scientific Research Fund) project entitled ***„Application of Concept Lattices and Fuzzy Methods in Group Technology"*** (Id. No.: T03243, headed by Tóth, T.). In addition, the research has also been supported by the Hungarian Academy of Sciences (HAS) within the framework of ***Production Information Engineering Research Team*** (PIERT) established at the Department of Information Engineering in 1999.

This work connected with the project **„Digital Factory"** therefore it has has also been supported within the framework of ***National Research & Development Program*** (NRDP) grant No. 2/040/2001.

The financial support of the research by the three sources is gratefully acknowledged.

REFERENCES

1. WILLIAMS, P.J.: *Computer Integrated Manufacturing (CIM): A Management Perspective.* Ryerson Polytechnical Institute CATE, Toronto, 1985.
2. *Group Technology and Manufacturing Cells.* In: Modern Machine Shop, NC/CIM Guidebook, 1988., p.230.
3. GALLAGHER, C.C. and KNIGHT, W.A.: *Group Technology Production Methods in Manufacture*. Ellis Horwood Ltd 1986.
4. RADELECZKI, S. and TÓTH, T.: *Planning Discrete Technology Processes by means of Artificial Intelligence Methods: Application of Fuzzy Methods in Group Technology.* Research Report, OTKA 2348/91., 1992. (in Hungarian).
5. XU, H. and WANG, H.P.: *Part Family Formation for GT Applications Based on Fuzzy Mathematics.* Int. J. Prod. Res. Vol 27., No.9., 1989., pp.1637-1651.