KNOWLEDGE-BASED SYSTEM FOR GROUP TECHNOLOGY (KBGT)

Andrew Kusiak and Wadood M. Ibrahim

Department of Mechanical and Industrial Engineering
The University of Manitoba
Winnipeg, Manitoba, R3T 2N2
Canada

ABSTRACT

In this paper a knowledge-based system (KBGT) for solving the group technology problem is presented. The formulation of the group technology problem involves constraints related to machine capacity, material handling system capability, and machine cell dimension. It has been developed for an automated manufacturing system. The KBGT takes advantage of the developments in expert systems and optimization. Two basic components of the knowledge-based system, namely the knowledge-based subsystem and the heuristic clustering algorithm are discussed. Each partial solution generated by the clustering algorithm is evaluated for feasibility by the knowledge-based subsystem which modifies search directions of the algorithm. The KBGT is illustrated with numerical examples.

1. INTRODUCTION

The concept of cellular manufacturing (known also as the group technology concept) is to decompose a manufacturing system into subsystems, which are easier to manage than the entire manufacturing system. Machines are grouped into machine cells (MC) and parts are grouped into part families (PF). The cellular concept is illustrated in Example 1 with a binary machine-part incidence matrix.

Example 1

Consider the machine-part incidence matrix (1)

$$[a_{ij}] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \text{PART NUMBER} \\ \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 1 & & & 1 \\ & & 1 & \\ 1 & & & 1 \\ & 1 & 1 & \end{array}\right] \end{array} \quad \text{MACHINE NUMBER} \qquad (1)$$

$$\text{where } a_{ij} = \begin{cases} 1 & \text{if machine } i \text{ is used to process part } j \\ 0 & \text{otherwise} \end{cases}$$

Rearranging rows and columns in (1) results in matrix (2).

$$\begin{array}{c|c|cccc|} & & \multicolumn{2}{c}{\text{PF-1}} & \multicolumn{2}{c}{\text{PF-2}} \\ & & 1 & 4 & 2 & 3 \\ \hline \text{MC-1} & 1 & 1 & 1 & & \\ & 3 & 1 & 1 & & \\ \hline \text{MC-2} & 4 & & & 1 & 1 \\ & 2 & & & & 1 \\ \hline \end{array} \qquad (2)$$

Two machine cells MC-1={1,3}, MC-2={4,2} and two corresponding part families PF-1={1,4}, PF-2={2,3} are visible in matrix (2). There is no flow of parts between the two machine cells. One has to realize, that it is virtually impossible to design a cellular manufacturing system without any interaction among machine cells. A typical situation occuring in practice is illustrated in matrix (3).

$$\begin{array}{cc|ccccc|} & & \multicolumn{5}{c}{\text{PART NUMBER}} \\ & & 2 & 5 & 3 & 1 & 4 \\ \hline \text{MC-1} & 2 & 1 & 1 & 1 & & \\ & 4 & 1 & 1 & & & \\ \hline \text{MC-2} & 1 & & & 1 & 1 & 1 \\ & 3 & & & 1 & 1 & \\ \hline \end{array} \quad \text{MACHINE NUMBER} \qquad (3)$$

Part 3 is to be manufactured in both machine cells.

The cellular concept should be applied only when the degree of interaction among different machine cells is low. One of the elements of cellular manufacturing is a cellular layout of machines discussed in [6], [14]. Readers interested in the survey of GT literature may refer to [19], where some 400 references have been listed. Introduction to group technology is presented in [5] and [7].

The problem of grouping machines and parts can be solved by clustering analysis which has been used in a much broader context. For review of application of clustering analysis see for example [1] and [4]. Since the clustering problem is NP-complete [16], heuristic algorithms are most likely to be used for large scale industrial problems. An industrial grouping problem solving may involve, for example, 200 machines (rows in matrix $[a_{ij}]$) and 2,000 parts (columns in matrix $[a_{ij}]$).

2. FORMULATION OF THE GROUPING PROBLEM IN AUTOMATED MANUFACTURING SYSTEMS

A typical approach to cellular manufacturing is to group machines and parts based on the binary machine-part incidence matrix, usually without any constraints. Some authors, for example [18], [9], [11] have restricted the size of machine cells and part families.

The approach presented in this paper considers two formulations of the grouping problem.

The first formulation is a generalization of the grouping problem presented in the literature. Rather than the binary matrix $[a_{ij}]$, matrix $[t_{ij}]$, where $t_{ij} \geq 0$ is the processing time of part j on machine i is considered. This formulation involves also some constraints, which are typical for an automated manufacturing environment.

The grouping problem in automated manufacturing systems can be loosely formulated as follows [12,14] :

Determine machine cells; for each machine cell, select a part family consisting of parts with maximum sum of production costs and select a suitable material handling carrier with the minimum corresponding cost subject to the following constraints :

Constraint C1 : processing time available at each machine is not exceeded

Constraint C2 : upper limit on the frequency of trips of material handling carriers for each machine cell is not exceeded

Constraint C3 : number of machines in each machine cell does not exceed its upper limit or

alternatively the dimension (for eg. the length) of each machine cell is not exceeded.

The above formulation of the GT problem is not only computationally complex, but also involves constraints which are difficult to handle by any algorithm alone. Therefore, to solve the the above problem, a knowledge-based system has been developed.

The second formulation considered is a special case of the generalized formulation of the group technology problem. It involves a 0-1 machine-part incidence matrix (see Example 1) and constraint C3 presented above.

3. STRUCTURE OF THE KNOWLEDGE-BASED SYSTEM

A typical knowledge-based system, (called also a stand alone knowledge-based system), is developed based on the knowledge elicited from experts. The elicited knowledge is represented using a suitable knowledge representation scheme in a knowledge base. A control strategy, implemented in a form of an inference engine, is employed to search the knowledge base in order to solve a problem. A knowledge-based system of this structure is suitable rather for qualitative problems, but is inefficient for solving problems of quantitative nature.

In this paper, a tandem knowledge-based system is considered, where a knowledge-based subsystem and an algorithm closely interact with each other. The algorithm deals with the quantitative aspects of the problem while the knowledge-based subsystem deals with the qualitative aspects. An earlier version of this system is presented in [13].

The knowledge-based system (KBGT) considered has the structure shown in Figure 1 .

The KBGT consists of five components :
(1) data base
(2) knowledge base
(3) inference engine
(4) request processor
(5) clustering algorithm.

Each of these components is discussed in detail in sections 4, 5, and 6.

One of the most tangible advantages of the tandem architecture is a relatively small knowledge base. This is because the computational effort is divided between the inference engine and the algorithm. For the same reason the tandem knowledge-based system is typically faster than a stand-alone system.

The KBGT has been implemented in Common LISP on a SPERRY-PC. LISP, as a programming language for KBGT, has been selected for three reasons:

(1) it facilitates implementation of the declarative knowledge
(2) it facilitates implementation of the procedural knowledge (the clustering algorithm)
(3) it provides flexibility to define and implement the interaction between the the algorithm and the knowledge-based subsystem.

3.1 Input data

The input data required by KBGT fall into two categories :
(i) machine data
(ii) part data

In addition to the above, depending on the characteristics of the manufacturing system, the following optional data can be provided :

(iii) maximum number of machines in a machine cell

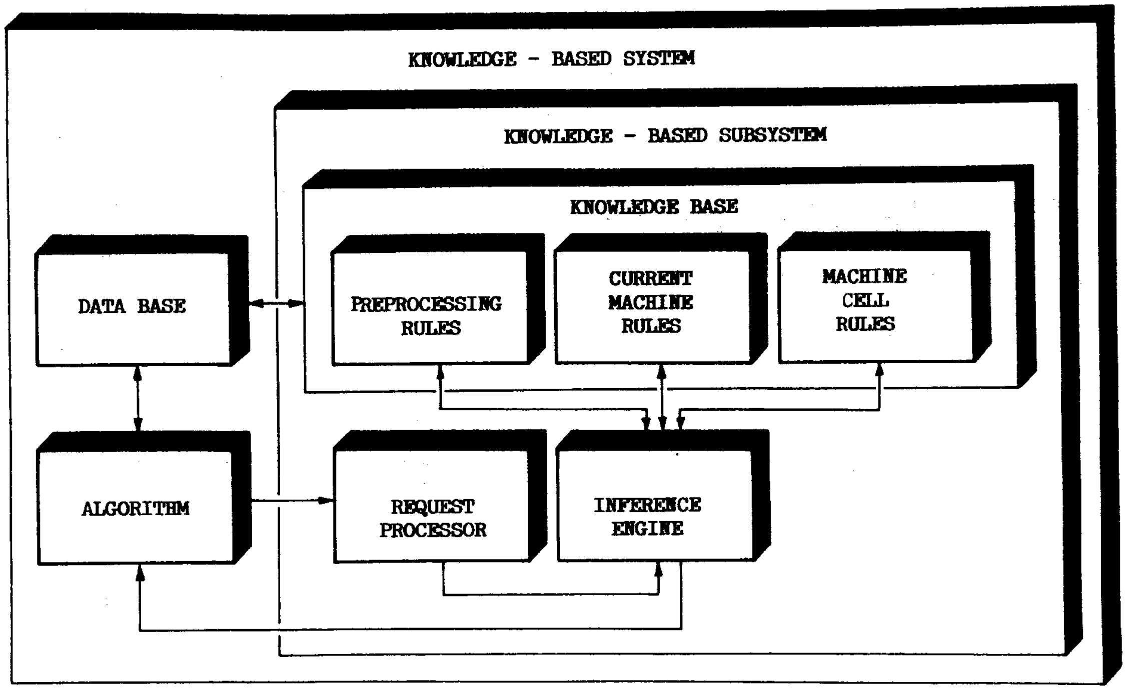

Figure 1. Structure of the knowledge-based system (KBGT)

(iv) maximum frequency of trips which can be handled by a material handling carrier (for example, robot or automated guided vehicle, AGV).

Details regarding the input data are discussed in section 4.

3.2 Grouping Process

Prior to begining of the grouping process, KBGT constructs a machine-part incidence matrix based on the input data provided by the user.

Next, the KBS initializes in the data base, objects representing facts known about the manufacturing system considered. Then the system forms machine cells and the corresponding part families. Each machine cell is formed by including one machine at a time. A machine is first analyzed by the KBS for the possibility of inclusion in the machine cell. For example, a bottleneck machine, i.e. machine that process parts visiting more than one machine cell are not included.

Each time a machine cell has been formed, the KBS checks whether any of the constraints C1-C3 has been violated and removes all parts violating the constraints.

For a machine cell, which has been formed and analyzed by the KBS, the corresponding machines and parts forming a part family are removed from the machine-part incidence matrix. The system does not backtrack in the grouping process, i.e. once a machine cell is formed, the machines included in the cell are not considered for future machine cells. This irrevocable control strategy, as illustrated in Figure 2, is possible due to the nature of the algorithm and the knowledge-based analysis performed by the KBS.

3.3 Output Data

At the end of the clustering process, KBGT prints the following data:

(1) machine cells formed
The machine cells formed are listed in the order they have been generated. For each machine cell the following information is provided :

- machine cell number
- list of machine numbers in a machine cell
- part family number
- list of part numbers in a part family
- material handling carrier alternatives, if any.

(2) part waiting list
This list includes parts that were placed on the waiting list due to either :

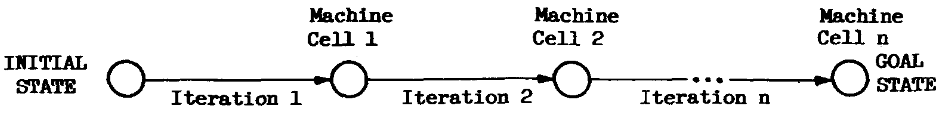

Figure 2. Illustration of the irrevocable control strategy of KBGT

- overlaping of parts in such a way that prevents grouping, or
- including them in a machine cell would violate one or more constraints.

(3) list of machines not used
The list of machines with all parts removed during the grouping process

(4) list of bottleneck machines
The list of machines that process a relatively large number of parts, which need to be processed on machines belonging to more than one machine cell. These machines should be given special consideration while determining the layout. Each of these machines should be preferably located adjacently to the machine cell that processes the same parts.

(5) maximum number of machines in a cell
This number indicates the maximum number of machines in a machine cell. It has an impact on the machine cells formed, namely if it was too small, it might result in the removal of too many parts. If this number is not supplied by the user then the system calculates it using a formula based on the number of machines in the manufacturing system considered.

4. DATA BASE

The global data base contains information about the current problem in a form of objects and frames. It is a non-monotonic data base, since objects are modified by the grouping algorithm and KBS.

The contents of the data base are either provided by the user as input data, or generated by the system. A list of objects and frames in the data base is as follows :

(1) machine frame
Machine frame contains information regarding one machine and is identified by the machine number. It has the following format :

```
(m#i  ((parts ((p#1  p-time)...(p#j  p-time)...))
       (max-process-time   x)
       (multiple  y)))
```

where :

p-time: the time required to process part number p#j on machine m#i (p-time is equal to 0, if processing time is not available)

max-process-time (optional) : maximum processing time available on machine m#i, i.e. the capacity of machine m#i

multiple (optional) : number of the identical machines available.

(2) part frame
Part frame contains information regarding each part and is identified by the part number. It has the following format :

```
(part#j  ((primary-pp  (m#1 ... m#i...))
          (fr   y)
          (fa   z)))
```

where :
primary-pp: primary process plan for part#i
fr : frequency of trips required by a robot to handle part#i
fa : frequency of trips required by an AGV to handle part#i

(3) machine-part incidence matrix
The machine-part incidence matrix is constructed by the system based on the input data provided in the following format :

```
( (m#1  ((p#1  p-time) (p#2  p-time)...))
   .
   .
   .
  (m#i  ...                              )
   .
   .
   .                                      )
```

In case when processing times are not available, then by default the matrix is a 0-1 incidence matrix, i.e. p-time is 0.

(4) current machine
A machine that the system has selected for possible inclusion in the machine cell being formed.

(5) list of candidate machines
A list of candidate machines to be included in the machine cell being formed.

(6) list of temporary candidate machines
A list of all machines such that the parts that are processed on the current machine are also processed on one or more of these machines. Moreover, these machines are not on the list of candidate machines.

(7) part waiting list
A list of parts that have been removed from the machine-part incidence matrix.

(8) list of bottleneck machines
A list of all bottleneck machines.

(9) list of temporary bottleneck machines
A list of machines that are considered to be temporary bottleneck machines. These machines may become non-bottleneck machines after some parts have been removed from the machine-part incidence matrix.

(10) list of machines not used
Same as discussed earlier in the subsection on the output of KBGT.

(11) MC-k (machine cell k)
A list of machine numbers in the current machine cell.

(12) PF-k (part family k)
A list of part numbers in the current part family.

5. THE KNOWLEDGE-BASED SUBSYSTEM (KBS)

As illustrated in Figure 2, the knowledge-based subsystem (KBS) consists of three components :
(1) knowledge base
(2) inference engine
(3) request processor.

5.1 Knowledge base

The knowledge base in KBGT contains production rules, which have been acquired from three experts in group technology. In the current implementation of KBGT, the knowledge base consists of three classes of production rules :
(a) preprocessing rules
(b) current machine rules
(c) machine cell rules.

The preprocessing rules deal with the initialization of objects in the data base that are not provided by the user. The current machine rules check the appropriateness of a current machine to the machine cell being formed, for example whether the current machine is a bottleneck machine. The machine cell rules deal with each machine cell which has been formed. Machine cell rules check for violation of constraints and remove parts violating them. The structuring of rules into separate classes has two advantages. First, the search for applicable rules is more efficient since the inference engine attempts to fire only rules that are relevant to the current context. Second, the modularity of the rule base makes it more understandable, and easy for modification.

Each rule has the following format :

(rule-number (IF conditions
THEN actions))

The rule number is used for identification by the inference engine. The most significant digit represents a class, and the other two digits represent a rule number in a class. The actions of the rule are carried out, only if the conditions in the IF part are true. Each condition in a rule can have one of the following three forms :
(i) a straightforward checking of values in the data base,
(ii) procedure calls to calculate the values required , or
(iii) a combination of (i) and (ii).

An example of (i) is comparing the size of the current MC-k with the maximum number of machines allowed per machine cell. An example of (ii) is a call of the procedure calculating the number of parts that a machine shares within the current MC-k. The action part consists of procedure calls and/or modifications of values of some objects in the data base.

Combining a rule-based representation paradigm with procedural representation paradigm, improves the efficiency of the KBS. Values that can not be obtained directly from the data base are calculated only when necessary.

Sample production rules that have been implemented in KBGT are listed below :

Rule-102 (preprocessing rule)

IF the maximum number, N, of machines in a machine cell is not specified

THEN calculate N as follows :
$N = (\lceil$ the total number of machines $/3 \rceil + 1)$,

where $\lceil \bullet \rceil$ denotes the integer part of $\bullet$.

Rule-103 (preprocessing rule)

IF the maximum number of machines in a machine cell is specified

THEN remove from the machine-part incidence matrix all parts that require a number of machines greater than the maximum number of machines in a machine cell
AND place them on the part waiting list.

Rule-201 (current machine rule)

IF no machines have been included in MC-k
AND the number of temporary candidate machines plus the current machine is greater than the maximum number of machines per machine cell

THEN add the current machine number to the list of temporary bottleneck machines
AND send a message to the algorithm to select another machine.

Rule-301 (cluster rule)

IF there are machines where constraint C1 is violated

THEN remove parts from the machines violating constraint C1
AND place the removed parts on the part waiting list.

5.2 Inference engine

One of the greatest advantages of the tandem system architecture is the simplicity of the inference engine. The inference engine in KBS employs a forward-chaining control strategy. In a given class of rules it attempts to fire all the rules which are related to the context considered. If a rule is triggered, i.e. the conditions are true, then the actions of the triggered rule are carried out. However, some rules, for example Rule-102, stop the search of the inference engine for applicable rules and to send a message to the algorithm.

The inference engine maintains a list of the rules which have been fired. This list is called "explain". The rules in "explain" are placed in the order that they were fired. The list forms a basis for building an explanation facility.

5.3 Request processor

The request processor facilitates the interaction between the algorithm and KBS. Based on each request of the algorithm, the request processor calls the inference engine and selects a suitable class of rules to be searched by the inference engine.

6. CLUSTERING ALGORITHM

The clustering algorithm presented is an extention of the algorithm presented in [15]. It takes advantage of two simple observations :

Observation 1

A horizontal line h_i drawn through any row i (machine number i) of matrix $[t_{ij}]$ indicates parts to be manufactured on machine i. This observation is illustrated in matrix (4)

$$
[t_{ij}] = \begin{array}{cc} & \text{PART NUMBER} \\ \text{MACHINE NUMBER} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.5 & 3.1 & & 2.8 & \\ & & 7.4 & & 1.6 \\ 3.9 & & 4.2 & & 4.3 \end{bmatrix} \begin{array}{c} \\ -h_2 \\ \\ \end{array} \end{array} \qquad (4)
$$

The horizontal line h_2 crosses elements (2,3) and (2,5) in matrix (4). Parts 3 and 5 are to be manufactured on machine 2.

Observation 2

A vertical line v_j drawn through any column of matrix $[t_{ij}]$ indicates machines to be used for manufacturing of the corresponding parts.

For any two 0-1 vectors

$$m_i = [a_{i1}, a_{i2}, \ldots, a_{ik}, \ldots, a_{in}]$$

$$m_j = [a_{j1}, a_{j2}, \ldots, a_{jk}, \ldots, a_{jn}]$$

define a similarity measure between machines i and j

$$s_{ij} = \sum_{k=1}^{n} \delta(a_{ik}, a_{jk})$$

where :

$$\delta(a_{ik}, a_{jk}) = \begin{cases} 1 & \text{if element } a_{ik} = a_{jk} \\ 0 & \text{otherwise} \end{cases}$$

n number of parts

Based on the two observations the clustering algorithm is developed.

6.1 The Algorithm

Step 0 : Set iteration number k=1 ;
Construct machine-part incidence matrix ;
Send a request to KBS for preprocessing.

Step 1 : Select a machine (row of machine-part incidence matrix) such that it processes the maximum number of parts and is not in the list of temporary bottleneck machines;
Place the selected machine in the list of candidate machines.

Step 2 : From the list of candidate machines, select a machine, which is the most similar to the current machine cell. If the current MC-k is empty, then choose the machine selected in Step 1 ;
Draw a horizontal line h_i, where i is the selected machine number.

Step 3 : For each entry crossed by the horizontal line h_i draw a vertical line v_j. Parts indicated by the vertical lines are potential candidates for the current part family PF-k ;
For each entry $t_{ij} > 0$ crossed by a vertical line v_j, add the corresponding machines to the list of temporary candidate machines ;
Remove the current machines from the list of candidate machines.

Step 4 : KBS analyzes the current machine selected, and takes one of following two actions :

- go to Step 5 (include the current machine in current MC-k)
- go to Step 1 (do not include it).

Step 5 : Add the machine considered to the current machine cell MC-k ;
Add the corresponding part numbers to the current PF-k ;
If the list of candidate machines is empty, then go to Step 6, otherwise, go to Step 2 .

Step 6 : KBS analyzes the current machine cell MC-k for violations of constraints C1-C3 and attempts to satisfy the constraints ;
Remove the current machine cell MC-k and the corresponding part family PF-k from the machine-part incidence matrix ;
Increment k by 1 .

Step 7 : If the machine-part incidence matrix is not empty, then go to Step 1 ; otherwise,

STOP.

7. APPLICATION OF KBGT

To illustrate and test performance of the KBGT a number of problems have been considered. The first is a generalized group technology problem represented by matrix (5) of processing times. The column on the right hand side of matrix (5) represents the maximum processing time available on each machine. The two vectors fa and fr on the top of matrix (5) represent the frequency of trips required by an AGV (fa) and a robot (fr) to handle each part. A "-" indicates that the corresponding part can not be handled by an AGV or a robot. The maximum frequency of trips to be handled by an AGV and a robot are max-fa and max-fr, respectively.

	01	02	03	04	05	06	07	08	09	10	11	12	
fa	11	30	2.5	-	6	10	-	6	7	15	18	14	max-fa (40)
fr	11	30	5	3	6	15	10	12	7	-	36	28	max-fr (100)
PART NUMBER / MACHINE NUMBER													
1		4		21				8					40
2	26		5			10							40
3			20		10					22		8	40
4		35					2	6					50
5	5					6			25				50
6		16		10			3				1		60
7					1					7		7	20

(5)

The KBGT input represention of matrix (5) is shown in Figure 4.

```
(setq machines-db '(

      (m1 ( (parts ((p2 4)(p4 21)(p8 8)))
            (max-process-time 40)))
      (m2 ( (parts ((p1 26)(p3 5)(p6 10)))
            (max-process-time  40)))
      (m3 ( (parts ((p3 20)(p5 10)(p10 22)(p12 8)))
            (max-process-time  40)))
      (m4 ( (parts ((p2 35)(p7 2)(p8 6)))
            (max-process-time  50)))
      (m5 ( (parts ((p1 5)(p6 6)(p9 25)))
            (max-process-time  50)))
      (m6 ( (parts ((p2 16)(p4 10)(p7 3)(p11 18)))
            (max-process-time  60)))
      (m7 ( (parts ((p5 1)(p10 7)(p12 7)))
            (max-process-time  20)))
                                                        ))

(setq parts-db '(

        (p1  ((primary-pp (m2 m5))
              (fr 11) (fa 11)))
        (p2  ((primary-pp (m1 m4 m6))
              (fr 30) (fa 30)))
        (p3  ((primary-pp (m2 m3))
              (fr 5)  (fa 2.5)))
        (p4  ((primary-pp (m1 m6))
              (fr 3)  (fa 0)))
        (p5  ((primary-pp (m3 m7))
              (fr 6)  (fa 6)
        (p6  ((primary-pp (m2 m5))
              (fr 15) (f 10)))
        (p7  ((primary-pp (m4 m6))
              (fr 10) (fa 0)))
        (p8  ((primary-pp (m1 m4))
              (fr 12) (fa 6)))
        (p9  ((primary-pp (m5))
              (fr 7)  (fa 7)))
        (p10 ((primary-pp (m3 m7))
              (fr 0)  (fa 15)))
        (p11 ((primary-pp (m6))
              (fr 36) (f 18)))
        (p12 ((primary-pp (m3 m7))
              (fr 28) (f 14)))                ))

 (setq max-mc-k-size  0)
 (setq max-fr 100)
 (setq max-fa  40)
```

Figure 4. Input of matrix (5) in KBGT format

The output from KBGT for matrix (5) is shown in Figure 5.

```
((MACHINE-CELL  1)     M3  M7)
((PART-FAMILY   1)     P5 P10 p12)
(M-H-S-ALTERNATIVE    (AGV))
     +++++++++++++++++++

((MACHINE-CELL  2)     M6 M1 M4)
((PART-FAMILY   2)     P2 P4 P7 P11 P8)
(M-H-S-ALTERNATIVE    (ROBOT))
     +++++++++++++++++++

((MACHINE-CELL  3)     M5  M2)
((PART-FAMILY   3)     P1 P6 P9)
(M-H-S-ALTERNATIVE    (ROBOT OR AGV))
     +++++++++++++++++++

--------------------------------------
(PART-WAITING-LIST===========>   (P3))
(MACHINES-NOT-USED===========>     0 )
(BOTTLENECK-MACHINES=========>     0 )
---------------------------------------
(MAXIMUM-MACHINE-CELL-SIZE-USED    3)
```

Figure 5. KBGT output for matrix (5)

As shown in Figure 5, three machine cells and part families have been generated. MC-1 is served by an AGV, MC-2 is served by a robot, and MC-3 can be served by a robot or an AGV. The overlapping part 3 is placed on the part waiting list. The computation was performed for the maximum cell size equal 3.

The second problem is a special case of the generalized GT problem. It is based on 0-1 machine-part incidence matrix as shown in Figure 6.

PART NUMBER

MACHINE NUMBER	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1																																					1					1	
2		1								1																		1				1					1	1		1		1	
3							1										1																	1	1	1							
4					1				1					1					1		1		1						1														
5					1			1	1					1	1	1			1		1		1						1				1								1		1
6	1	1				1	1	1				1	1	1			1		1				1									1	1	1			1		1	1		1	1
7	1												1												1																		
8	1	1	1					1	1		1	1			1				1	1	1		1	1			1	1			1						1	1			1		1
9		1		1						1								1										1				1					1	1		1		1	
10	1											1	1												1	1					1								1				
11			1						1											1				1			1			1													
12											1											1		1			1			1													
13			1																					1																			
14		1				1											1																		1								
15					1									1					1			1											1								1		1
16		1					1			1								1														1					1	1				1	

Figure 6. Machine-part incidence matrix [2]

For the incidence matrix in Figure 6, the KBGT provides the solution in Figure 7.

To date a large number of clustering methods have been developed mostly for solving the 0-1 group technology problem and only a few of them have been tested.

Since the generalized formulation of the GT problem is new, we could not compare performance of the KBGT for this problem. We have identified four 0-1 problems in the GT literature and solved them with KBGT. The solutions obtained are of better quality than ones generated by the four algorithms considered (see Table 1). The computational time complexity of the heuristic clustering algorithms available in the literature is high, for example $O(m^2n+n^2m)$, where m is the number of rows and n is the number of columns in a machine-part incidence matrix (see [11]). The algorithm presented in this paper is an extention of the clustering identification algorithm [15] of the computational time complexity slightly higher than $O(mn)$. The CPU time reported in Table 1 is for a SPERRY-PC (an IBM-PC compatible). In addition some of the traditional algorithms listed in Table 1 required human interaction, while KBGT does not. The machine-part incidence matrices for each of the four problems presented in Table 1.

8. CONCLUSION

In this paper a generalized formulation of the problem of grouping machines and parts in an automated manufacturing system was presented. The formulation involves a matrix of processing times and three constraints related to the availability of processing time at each machine, requirement for material handling carriers and the maximum number of machines allowed per machine cell. A special case of the grouping problem formulated was also considered. To solve the grouping problem a knowledge-based system (KBGT) was developed. The KBGT involves a heuristic algorithm and a knowledge-based subsystem. To demonstrate performance of the knowledge-based system four problems available in the literature have been solved. The solutions obtained are superior to ones presented in the literature. This is due to the clustering algorithm presented and the group technology knowledge included in the knowledge base. The approach presented allows one to take advantage of user's production expertise. It can be applied for solving other problems as well.

```
((MACHINE-CELL  1)  M5 M4 M15)
((PART-FAMILY   1)  P5 P8 P14 P15 P16 P19 P21
                    P23 P29 P33 P41 P43)
     +++++++++++++++++++

((MACHINE-CELL  2)  M9 M2 M16 M1 M14 M3)
((PART-FAMILY   2)  P2 P4 P10 P18 P28 P32 P37
                    P38 P40 P6 P7 P17 P34
                    P35 P36 P42)
     +++++++++++++++++++

((MACHINE-CELL  3)  M10 M7)
((PART-FAMILY   3)  P1 P12 P13 P25 P26 P31 P39)
     +++++++++++++++++++

((MACHINE-CELL  4)  M12 M11 M13)
((PART-FAMILY   4)  P11 P22 P24 P27 P30 P3 P20)
     +++++++++++++++++++

---------------------------------------------
(PART-WAITING-LIST==========>  (P9))
(MACHINES-NOT-USED===========>  0 )
(BOTTLENECK-MACHINES=========> M8  M6)
---------------------------------------------
(MAXIMUM-MACHINE-CELL-SIZE-USED     6)
```

Figure 7. KBGT output for the machine-part incidence matrix in Figure 6

Table 1. Solutions of four group technology problems

PROBLEM			SOLUTION				
Problem Reference	Number of Machines	Number of Parts	Solution Method	Number of Machine Cells and Part Families	Number of Overlapping Parts	Bottleneck Machines	KBGT CPU Time
King [8]	16	43	ROC2 Algorithm[1]				
			• solution 1	–	–	2	
			• solution 2	4	3	2	
			KBGT	4	1	2	4 sec.
Seifoddini [17]	5	12	SLCA	2	5	0	
			KBGT	2	4	0	1 sec.
Kumar and Vannelli [10]	30	41	Kumar and Vannelli Algorithm				
			• solution 1	2	4	0	
			• solution 2	2	5	0	
			• solution 3	3	6	0	
			KBGT[2]	3	7	0	5 sec.
Chandrasekharan and Rajagopalan [3]	40	100	ZODIAC	10	33	0	
			KBGT[3]				
			• solution 1	5	23	0	23 sec.
			• solution 2	6	32	0	24 sec.
			• solution 3	7	31	0	40 sec.

1 solution 1 has been generated by the ROC2 algorithm while solution 2 is a visually modified solution

2 if the overlapping part 38 is subcontracted then machine 24 is redundant

3 solution 1, 2, and 3 have been obtained for the maximum machine cell size of 11, 10, and 9, respectively

REFERENCES

[1] ANDENBERG, M.R. 1973, "Cluster Analysis for Applications", (New York: Academic Press).

[2] BURBIDGE, J. L. 1973, "Production Flow Analysis on the Computer", The Institute of Production Engineers, Group Technology Division, Third Annual Conference.

[3] CHANDRASEKHARAN, M.P. and RAJAGOPALAN, R. 1987, "ZODIAC-an algorithm for concurrent formation of part-families and machine-cells", International Journal of Production Research, Vol. 25, No.6, pp. 835-850.

[4] EVERITT, B., 1980, "Cluster Analysis", (New York, Halsted Press).

[5] HAM, I. HITOMI, K. and YOSHIDA, T. 1985, "Group Technology", (Boston, MA: Kluwer-Nijhoff Publishing).

[6] HERAGU, S. and KUSIAK, A., 1988, "Machine layout problem in flexible manufacturing systems", operations Research, Vol. 36, No. 2.

[7] HYER, N.L. Ed., 1984, "Group Technology at Work", (Dearborn, MI: Society of Manufacturing Engineers).

[8] KING, J.R., and NAKORNCHAI, V., 1982, "Machine-component group formation in group technology: Review and extention", International Journal on Production Research, Vol. 20, pp. 117 133.

[9] KUMAR, K. R., KUSIAK, A. and VANNELLI, A. 1986, "Grouping of parts and components in flexible manufacturing systems", European Journal of Operational Research, Vol. 24, pp. 387-397.

[10] KUMAR, K. R., and VANNELLI, A. 1987, "Strategic subcontracting for efficient disaggregated manufacturing", International Journal of Production Research, Vol. 25, No. 12, pp. 1715-1728.

[11] KUSIAK, A., 1985, "The part families problem in flexible manufacturing systems. Annals of Operations Research, Vol. 3, pp.279-300.

[12] KUSIAK, A., 1987, "Artificial intelligence and operations research in flexible manufacturing systems", Information Systems and Operational Research (INFOR), Vol. 25, pp. 2-12.

[13] KUSIAK, A., 1988, "EXGT-S : A knowledge-based system for group technology", International Journal of Production Research, Vol. 26, No. 5.

[14] KUSIAK, A., 1988, "Artificial Intelligence : Implications for CIM, (New York: Springer-Verlag)

[15] KUSIAK, A., and CHOW, W.S., 1987, "Efficient solving of the group technology problem", Journal of Manufacturing Systems, Vol. 6, pp. 117-124.

[16] LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A.H.G., and SHOMYS, D.B. Eds., 1985, "The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization, (New York: Wiley)

[17] SEIFODDINI, H. 1986, "Improper Machine Assignment in Machine-Component Grouping in Group Technology", Proceedings of the Fall Industrial Engineering Conference, Boston, MA, December 7-10, pp. 406-409.

[18] STANFEL, L.E., 1982, "An algorithm using Lagrangean relaxation and column generation for one-dimensional clustering problems", in S.H. Zanakis and J.S. Rustagi (Eds.), Optimization in Statistics, (Amsterdam: North Holland), pp. 165-185.

[19] WAGHODEKAR, P.H., and SAHU, S., 1983, "Group technology: A research bibliography", OPSEARCH, Vol. 29, pp. 225-249.