Production Systems and Information Engineering
Volume 3 (2006), pp. 139-155

NEW APPROACHES IN SOLVING MACHINE-PART GROUPING PROBLEMS

TIBOR TÓTH
University of Miskolc, Hungary
Department of Information Engineering
toth@ait.iit.uni-miskolc.hu

ATTILA KÖREI
University of Miskolc, Hungary
Department of Information Engineering
matka@gold.uni-miskolc.hu

[Received March 2006 and accepted May 2006]

Abstract. Machine-part grouping problems arise in a production plant when forming a new production system or reorganizing an existing one. Application of Group Technology principles can help in finding the optimal layout and manufacturing system. In order to satisfy the basic principle *similar things should be done similarly* parts are assigned to different families based on their processing requirements and machines are separated into groups to process specific part families. The machine-part cell formation problem is a widely researched area and numerous algorithms have been developed to solve it. This paper provides a survey of the latest results related to clustering methods, artificial intelligence approaches and some mathematical techniques. In addition an abstract algebraic method based on the theory of concept lattices is also outlined.

Keywords: group technology, cell formation problem, similarity coefficients, genetic algorithms, mathematical programming, concept lattices.

1. Introduction

The philosophy of group technology (GT) plays an important role in the design of manufacturing cells. The basic concept of GT is to identify and exploit the similarity between parts, machines and manufacturing processes. GT is a disciplined approach to grouping items by their attributes. Parts having similar processing requirements are arranged into part families, and the machines processing them are grouped into cells. The advantages of applying group technology principles are reduced setup time, queuing time and material handling time, shorter lead times, reduced tool requirements and improved product quality.

The adoption of GT concepts yields an efficient production system and a significant reduction can be expected in overall manufacturing costs.

Cellular manufacturing (CM) can be regarded as one of the major applications of group technology. CM requires the identification of groups of similar parts and the associated machines which form cells. The determination of part families and machine cells is called the cell formation (CF) problem.

During the last three decades the CF problem has been widely researched and numerous methods have been developed for solving it. These methods are classified by several review papers: we follow the taxonomy proposed by Shafer [16]. The main aspect of his classification is the methodology the cell formation procedures are based on. From studying these methodologies, cell formation techniques can be arranged into the following six groups:

manual methods,

classification and coding approaches,

algorithms for sorting machine component matrix,

statistical cluster analysis,

artificial intelligence methods,

mathematical techniques.

In this paper we give a short survey of the latest results in solving CF problems. Some of these methods can be regarded as new approaches and others are improved versions of an older technique. We focus on the last three group of Shafer's taxonomy, and in the next three sections the main aspects and the latest results from these fields are reviewed. In Section 5 a new mathematical approach is described. This method is based on Formal Concept Analysis which is a prospering field of applied lattice theory.

2. Statistical Cluster Analysis

Application of a clustering technique requires the development of a measure quantifying the similarity or dissimilarity between two objects (parts or machines). Using the appropriate similarity measure the necessary production data can be incorporated in the early stages of the machine-part grouping procedure. Some of these factors are operation sequences, within-cell machine sequences, processing requirements of parts, production volumes, unit operation time, alternative process routings, etc. A number of similarity coefficients have been developed for taking into consideration the different production factors and goals during the CF process. The calculation of these coefficients combined with a clustering algorithm is called

a similarity coefficient based clustering method (SCM) and generally it consists of the following three steps:

a) Form the initial machine-part incidence matrix $[a_{ij}]$, where an entry "1" ("0") indicates that machine i is used (not used) to process part j. (Here i is the machine index (i=1,2,...,m), and j is the part index (j=1,2,...,n).
b) Choose a similarity coefficient and calculate for each pair of parts or machines the corresponding values. These numbers are stored in a similarity matrix whose elements represent the sameness between two parts or machines.
c) Based on the values of the similarity matrix a clustering algorithm expands the hierarchy of similarities among all pairs of parts (machines). Using the obtained tree or dendogram the part families (machine groups) can be identified.

Before applying some clustering technique we have to select a similarity coefficient which indicates the degree of similarity between object pairs. The most frequently employed coefficient is the Jaccard similarity coefficient which is defined for parts i and k in the following way:

$$J_{ik} = \frac{N_{ik}}{N_i + N_k - N_{ik}},$$

where N_{ik} is the number of machines that parts i and k have in common in their production, and N_i, N_k mean the number of machines processing part i and k respectively. Besides the Jaccard similarity coefficient numerous other similarity coefficients have been proposed in the literature. One of the most comprehensive reviews of the topic is given by Yin and Yasuda [21], who developed a new taxonomy to classify the various similarity and dissimilarity coefficients. Besides this classification they attempted to explain why similarity coefficient based methods are more flexible than other approaches. First, similarity coefficient methods apply cluster analysis to CF procedures. Clustering techniques are fundamental methods in group technology, being a basic approach for estimating similarities. Fitting well to the main idea of GT, a similarity coefficient based method can be a more effective way to solve CF problems. Another reason for the preference of SCM methods is that they are more suitable for certain principles which are generally accepted in solving complex problems:

(i) decomposition of the problem into small conquerable problems and
(ii) decomposition of the solution into small tractable stages.

These principles are satisfied by a clustering method because it consists of three steps as mentioned in the beginning of this section. These steps are independent of

each other, which makes it possible to reselect the similarity coefficients when extending the problem to incorporate additional production factors.

It is worth mentioning that the use of similarity coefficients is often combined with other techniques, mostly artificial intelligence methods for solving machine-part grouping problems. For example Tóth and Molnár [17] have developed two algorithms for forming part-groups and inserting new parts in existing groups. Starting from the similarity matrix of the parts they used fuzzy classification to solve the problems. In [10], Jeon and Leep proposed a new similarity coefficient, which considers alternative routes, and based on these coefficients the part families are identified by using genetic algorithm. Adenso-Díaz et al. in [1] suggested weighted similarity coefficients and Tabu search for determining machine cells.

3. Artificial Intelligence Methods

With the increasing speed and capacity of today's computers, researchers frequently apply artificial intelligence methods to solve the CF problem. The most commonly used techniques are pattern recognition, fuzzy reasoning, neural networks and genetic algorithms. Another reason for successful application of these methods is the NP-completeness of the cell formation problem, i.e. there is no algorithm of polynomial complexity to solve it. This means that methods using heuristics can be more suitable to solve the CF problem than other exact approaches.

In this section we focus on the methods using genetic algorithms. The basic ideas of these methods are discussed and some recent results are referred to.

Genetic algorithm (GA) is a heuristic search technique which was introduced by Holland in 1975 [9]. It is based on an analogy to natural selection and Darwin's evolution concepts. First a chromosome structure is to be defined to represent the solutions of the optimisation problem. After generating an initial solution population (which is done mostly randomly) some members of the population are selected to be parents to produce offspring. The selection is based on the so-called fitness function: the higher an individual's fitness value the more likely that individual is to be selected, satisfying the principle of survival of the fittest. The less fit members are replaced by new ones, who are produced by the parents using genetic operators: crossover and mutation. Crossover combines the best parts of parent chromosomes in order to exploit promising areas of the search place. Mutation is a small random modification of the chromosome that increases the diversity of the population and explores new regions of the search place. The process is repeated until a termination criterion is reached.

The main questions before implementing genetic algorithm are the following:

how to encode the structure of the chromosomes for representing solutions,
how to generate the initial population,
how to choose a good fitness function,
how to define the genetic operators,
how to choose the parameters according to the crossover and mutation operator, the population size, rate of individuals to be selected, and maximum number of iterations.

There are several possibilities for encoding chromosomes. Most of the studies use an integer codification to represent solutions. For example in a machine cell formation problem the following chain

m_1	m_2	m_3	m_4	m_5
1	2	1	2	1

represents a solution where two cells are formed, the first contains the machines m_1, m_3, m_5 while machines m_2, m_4 belong to the second cell. The main drawback of this representation that it induces redundancy since the cell indices can be permuted, so the chromosome

m_1	m_2	m_3	m_4	m_5
2	1	2	1	2

represents the same solution as the previous one. The redundancy grows very quickly with the number of cells, making the search for good solutions even more difficult. Boulif and Atif [2] avoided this difficulty by choosing binary coding for the chromosomes This method is based on the graph theory model of the CF problem, where the nodes represent machines and an edge of the graph indicates whether there is inter-machine traffic between the two vertices of this edge. An edge is encoded by 0 if the traffic between its two vertices is intracellular (expressing that the machines corresponding to the two vertices are in the same cell). The intercellular edges are denoted by 1. Using this reduced alphabet the GA algorithm can be implemented in an efficient way, and the search for good solutions becomes easier.

Although genetic algorithm is one the most popular methods for solving CF problems, we have to mention some disadvantages of this technique. The main problem with the standard GA approach is its weakness, which means that it does not incorporate problem-specific knowledge. The other drawback is related to the standard encoding scheme (both with integer and binary coding); which can cause unexpected effects when applying GA operators. To overcome these problems De Lit, Falkenauer and Delchambre propose a grouping genetic algorithm (GGA) to

solve the cell formation problem [6]. GGAs are a special class of genetic algorithms introduced by Falkenauer in 1992 [7] modifying the standard GAs to better match the structure of grouping problems. There are two main differences between GGA and the classic genetic algorithm: GGA uses a group oriented encoding scheme and special genetic operators suitable for the chromosomes. For solving a machine-part cell formation problem the GGA applies the following chromosome representation:

$$p_1 p_2 p_3 ... p_P \mid m_1 m_2 m_3 ... m_M \mid g_1 g_2 g_3 ... g_G$$

where p_i is the group to which part i is assigned, m_j is the group to which machine j is assigned and g_k is an existing group number. P denotes the number of parts, M denotes the number of machines in the problem and G is the number of groups in the solution. The main characteristic of the genetic operators of the GGA is that they work with the group part of the chromosomes rather than items [6]. In [3], Brown and Sumichrast compared the performance of a GGA against the performance of a standard GA approach in three different grouping problems. Their second problem was the machine-part cell formation problem and they used grouping efficacy to compare solution quality. Grouping efficacy can be computed by the following formula:

$$GE = 1 - \frac{e_0 + e_v}{e + e_v},$$

where e is the total number of ones in the original machine-part incidence matrix, e_v is the number of voids and e_0 is the number of exceptional elements in the solution. (Voids occur when a part does not require one of the machines in its group and exceptional elements arise when a part requires a machine from another group.) Using this performance measure the authors tested both solution techniques and they concluded that GGA outperformed the standard genetic algorithm in solving the machine-part cell formation problem, and is indeed an efficient technique even for large-sized problems.

4. Mathemathical Techniques

Mathematical techniques include methods related to graph theory, combinatorial analysis and mathematical programming. One of the widely researched exact methods is the integer programming approach. In group technology Kusiak was the first who adopted a linear programming method for part-family formation [12]. The suggested p-median model uses n^2 decision variables as follows:

$$y_{ij}= \begin{cases} 1, \text{ if part } i \text{ belongs to part family } j \\ 0, \text{ otherwise} \end{cases}$$

$i=1,2,...,n$, where n is the number of parts. Denote the desired number of part-families by p and compute the similarity coefficients s_{ij} between parts i and j in the following way:

$$s_{ij}^{(p)} = \sum_{k=1}^{m} \delta(a_{ki}, a_{kj}) \qquad i \neq j, \qquad i, j=1,2,...,n.$$

$$s_{ii}^{(p)} = 0 \qquad\qquad i=1,2,...,n.$$

Here a_{ki} is the element of the kth row and ith column of the machine-part incidence matrix and δ is the Kronecker function: $\delta(a_{ki}, a_{kj})=1$ if $a_{ki} = a_{kj}$, and 0 otherwise.

The objective is to maximize the sum

$$\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}^{(p)} x_{ij}$$

satisfying the following conditions:

(1) $\sum_{j=1}^{n} y_{ij} = 1$ for all $i = 1,...,n$,

(2) $\sum_{j=1}^{n} y_{jj} = p$,

(3) $y_{ij} \leq y_{jj}$ for all $i = 1,...,n, \ j = 1,...,n$

Condition (1) ensures that each part belongs to exactly one part family, and (2) specifies the required number of part families. Constraint (3) ensures that part i is grouped into the part family represented by j, if this family exists.

The part family formation method described here can be readily adapted to form machine cells first. In this case we have to compute similarity coefficients between machines, for example we can use the definition suggested by Wei and Kern [19]:

$$s_{ij}^{(m)} = \sum_{k=1}^{n} \Gamma(a_{ik}, a_{jk}), \text{ where } \Gamma(a_{ik}, a_{jk}) = \begin{cases} n\text{-}1 & \text{if } a_{ik} = a_{jk} = 1, \\ 1 & \text{if } a_{ik} = a_{jk} = 0, \\ 0 & \text{if } a_{ik} \neq a_{jk}. \end{cases}$$

These coefficients are applied by Won and Lee [20], who suggested two modified versions of the *p*-median model. They started from an extended *p*-median model, where the decision variables are the following:

$$x_{ij} = \begin{cases} 1, \text{ if machine } i \text{ is clustered into cell } j \\ 0 \text{ otherwise,} \end{cases}$$

and the objective function is $\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}^{(m)} x_{ij} \longrightarrow \max,$

subject to

$$(4)\ \sum_{i=1}^{m} x_{ij} = 1 \text{ for all } j = 1,...,m,$$

$$(5)\ \sum_{j=1}^{m} x_{jj} = p,$$

$$(6)\ \sum_{j=1}^{m} x_{ij} \geq L x_{jj} \text{ for all } i = 1,...,m,$$

$$(7)\ \sum_{j=1}^{m} x_{ij} \leq U x_{jj} \text{ for all } i = 1,...,m.$$

Conditions (4) and (5), similarly to the original formulation, ensure that each machine is assigned exactly to one cell and the number of machine cells is prescribed. In constraints (6) and (7) the number of machines grouped in to cell *i* is limited: at least *L* machines should be assigned to cell *i* only if cell *i* is formed, and *U* is the maximum number of machines allowed in each cell.

Since in most practical problems parts outnumber machines ($n>m$), the extended formulation with its m^2 binary variables leads to a smaller linear integer

programming problem compared to Kusiak's model. In spite of this reduction the extended model needs further improvement because of the difficulties in its implementation. The problem is how to choose the optimal median number p. In order to avoid this problem the entire model is tested for p=2,...,m, and then the best solution is selected. This type of implementation can hardly be carried out, because running the entire model for varying values of p causes too much computation time even on a medium sized CF problem. To overcome these difficulties Won and Lee introduced a special set of machines that have a high probability of serving as medians or seed machines for clustering. They developed an algorithm for determining the candidate set of median machines and after this with the modification of constraints, speedier implementation was achieved by excluding a large amount of binary variables.

In addition Won and Lee proposed another modified formulation of the model which makes further reduction of the number of variables possible and they presented remarkable test results, applying their method on large-sized CF problems. (The applications of integer programming approaches in CF problems containing 40 or more machines are rarely reported in the literature because these methods have so far required enormous computation time.)

Another opportunity to modify the original p-median model is the linear assignment model proposed by Wang [18]. The basic idea of the group formation algorithm is the selection of the p most dissimilar parts or machines. These items probably will be assigned to different groups since they have dissimilar design or manufacturing features. These group representatives can be determined recursively by using the similarity coefficients. With the knowledge of the group representatives the model can be formulated both for part family and machine cell formation and it contains far fewer decision variables compared with the p-median model (pn instead of n^2 or pm instead of m^2). The reduction in number of variables encouraged Wang to test his method on medium sized CF problems (the maximal number of machines was 40) and it proved an efficient method in a comparative study [18].

5. Solving Cell Formation Problems with Concept Lattices

In this section a new mathematical approach for solving machine-part cell formation problem is presented. The method has been developed at the University of Miskolc, in a collaborative project between the Department of Information Engineering and the Institute of Mathematics. This abstract algebraic method is related to Formal Concept Analysis, which can be regarded as a field of applied lattice theory. For the sake of completeness the basic elements of Formal Concept

Analysis are briefly introduced: for details see the fundamental work of Ganter and Wille [8].

The theory of Formal Concept Analysis is based on the theory of complete lattices. An ordered set $L=(L,\leq)$ is called a *lattice* if for any two elements x and y the supremum $x\vee y$ and the infimum $x\wedge y$ always exist. L is called a *complete lattice* if the supremum $\vee S$ and the infimum $\wedge S$ exist for any subset S of L. Every complete lattice has a largest (unit) element and a smallest (zero) element. The elements a and b are *neighbours* if $a<b$ and there is no element c fulfilling $a<c<b$ This relation is denoted by $a\prec b$ The neighbours of the zero elements are called the *atoms* of the lattice. A complete lattice in which every element is the supremum of atoms is called an *atomistic lattice*.

One of the basic notions of formal concept analysis is the term of formal context. A *formal context* $K=(G,M,I)$ consists of two sets G and M and a relation I between them. The elements of G are called the objects and the elements of M are called the attributes of the context (these traditional notations come from the German words *Gegenstand* and *Merkmal)*. The binary relation $I\subseteq G\times M$ is defined as follows: $(g,m)\in I$ if and only if the object $g\in G$ has the attribute $m\in M$. A small context can be represented by a cross table.

Observe that a machine-part incidence matrix can be considered as an analogous structure to the formal context. A given machine-part grouping problem corresponds to the formal context (G,M,I) where G is the set of machines, M contains the parts and I is determined by the incidence matrix with the following relation: $(g,m)\in I$ if the part m visits the machine g (Table 1).

For a set $A\subseteq G$ we define the set of the common attributes for the objects belonging to A: $A'=\{m\in M\mid (g,m)\in I,\ \forall g\in A\}$. Correspondingly for a set $B\subseteq M$ we define the set of the objects possessing all the attributes in B: $B'=\{g\in G\mid (g,m)\in I,\ \forall m\in B\}$.

Let $A\subseteq G$ be a set of objects and $B\subseteq M$ be a set of attributes. The pair $C=(A,B)$ is called a *formal concept* of the context (G,M,I) if the conditions $A'=B$ and $B'=A$ hold true. In this case A is called the *extent* and B is called the *intent* of the concept C with the notation $A=Ext(C)$ and $B=Int(C)$. For example ({1,4,5,10}, {a,b,c,o,p}) is a concept of the context K represented in Table 1.

Table 1. A machine-part incidence matrix with 11 machines and 22 parts. It corresponds to the formal context $K=(G,M,I)$, where G contains 11 objects, M contains 22 attributes and I is determined by the 1-s in the table

K		Parts																				
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
Machines	1	1	1	1								1				1	1				1	1	1
	2					1			1			1	1							1			
	3					1							1	1						1			
	4	1	1	1								1				1	1				1	1	1
	5	1	1	1				1								1	1				1	1	1
	6								1				1							1			
	7				1	1				1					1			1	1				
	8					1				1	1		1	1	1			1	1	1			
	9				1					1					1			1	1				
	10	1	1	1				1	1			1	1			1	1		1				
	11				1					1	1							1	1				

Denote by $L(G,M,I)$ the set of all concepts of the context (G,M,I) and introduce a partial order between the elements of it: $(A_1,B_1)\leq(A_2,B_2)$ if and only if $A_1 \subseteq A_2$ (then $B_2 \subseteq B_1$ also holds true). It can be verified that the lattice $(L(G,M,I),\leq)$ is a complete lattice and it is called the *concept lattice* of the context (G,M,I). The lattice $L(G,M,I)$ can be represented by a Hasse diagram, using the notion of neighbourhood. The elements of the lattice are depicted by circles in the plane. If x and y are concepts with $x \prec y$, the circle corresponding to x and the circle representing y are joined by a line segment. >From such a diagram the order relation can be read off as follows: $x < y$ if and only if the circle representing y can be reached by an ascending path from the circle representing x. The diagram of the concept lattice originating from the context K is presented in Fig. 1. The concept $c_1 = \{G,\varnothing\}$ represents the unit element of the concept lattice, its extent is equal to the full set of the objects, while the zero element is $c_0 = \{\varnothing,M\}$.

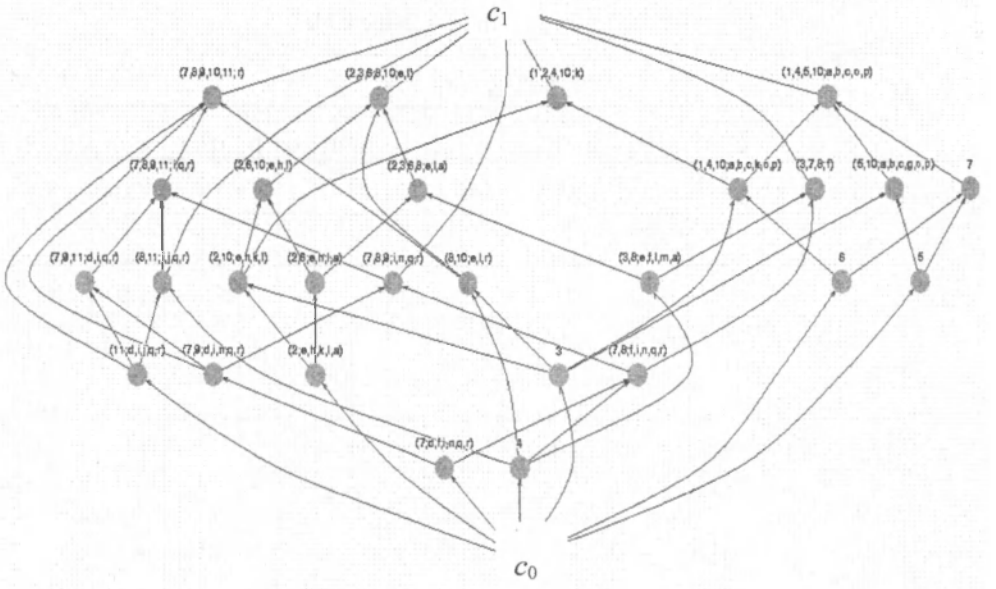

Figure 1. The concept lattice of the context K

A *classification system* S of the concept lattice $L(G,M,I)$ can be defined as a system of concepts where the extents of the concepts give a partition of the object set G. Formulating this definition we obtain

$$S = \{ (A_i, B_i) \mid i \in I \}$$

where $(A_i, B_i) \in L(G,M,I)$, $G = \bigcup A_i$, $A_i \cap A_j = 0$ if $i \neq j$.

For example the concepts ({2,3,6,8},{e,l,s}), ({1,4,5,10},{a,b,c,o,p}), ({7,9,11}, {d,i,q,r}) form a classification system of the concept lattice in Fig. 1.

The set of all classification systems of $L(G,M,I)$ is denoted by $Cls(L)$. We define an ordering relation between classification systems: $S_1 < S_2$ if the partition induced by S_1 refines the partition induced by S_2. It can be proved that the pair $(Cls(L), <)$ is a complete lattice which is called the *classification lattice* of $L(G,M,I)$. The 0-element of $Cls(L)$ is denoted by S_0, that is $S_0 = \wedge\{S : S \in Cls(L)\}$.

If the initial context consists of too many objects and attributes we obtain a large sized concept lattice and it is not simple to select those concepts from it that form a classification system. In this case using the results of Radeleczki [14] we can

determine the classification systems by means of a box lattice, which is a simpler structure than the original concept lattice.

The 0-element of $L(G,M,I)$ and any elements of a classification system are called *box elements* of $L(G,M,I)$. The set of the box elements are denoted briefly by $B(L)$. Restricting the partial order given on $L(G,M,I)$ to the set $B(L)$ we obtain a lattice again, the *box lattice* of $L(G,M,I)$. It can be verified that $(B(L),\leq)$ is an atomistic complete lattice, and its atoms are the elements of the finest classification system S_0. The box lattice originating from context K is shown in Fig 2.

The determination of the classification systems is carried out by means of box lattices in the following steps (more detailed discussion can be found in [11]).

a) Having started from the given context we determine the atoms of the box lattice.
b) We generate the further box elements using the observation that in a box lattice every element is a supremum of atoms.
c) Choosing the maximal disjoint systems of the box lattice we get the required classification systems (a maximal disjoint system cannot be extended by further box elements saving the property that the intersection of any two elements is the zero element).

Note that the determination of the classification systems gives a basis for formation of machine cells. The elements of a machine cell are usually characterized by the common parts processed by them. This means, using the notion of formal concept analysis, that $G_i''=G_i$ for every $i\in I$, where the sets G_i give a partition of the machine set G. In this case every block G_i is the extent of some concept of the concept lattice $L(G,M,I)$, because the pairs (G_i, G_i') satisfy the equations defining a concept. In other words the formation of machine cells can be solved by finding the suitable classification systems of $L(G,M,I)$.

Following the steps described above we have determined all of the classification systems of the concept lattice represented in Fig. 1. After applying step a) the following 7 atoms are obtained (the concepts are identified by their extents):

$a_1 = \{1,4\}$, $a_2 = \{5\}$, $a_3 = \{2,6\}$, $a_4 = \{3,8\}$, $a_5 = \{7,9\}$, $a_6 = \{10\}$, $a_7 = \{11\}$

The further box elements were determined by step b):

$d_1 = \{1,4,5,10\}$, $d_2 = \{7,9,11\}$, $d_3 = \{2,3,6,8,10\}$, $d_4 = \{5,10\}$, $d_5 = \{2,6,10\}$,

$d_6 = \{1,4,10\}$, $d_7 = \{2,3,6,8\}$, $d_8 = \{1,4,5\}$.

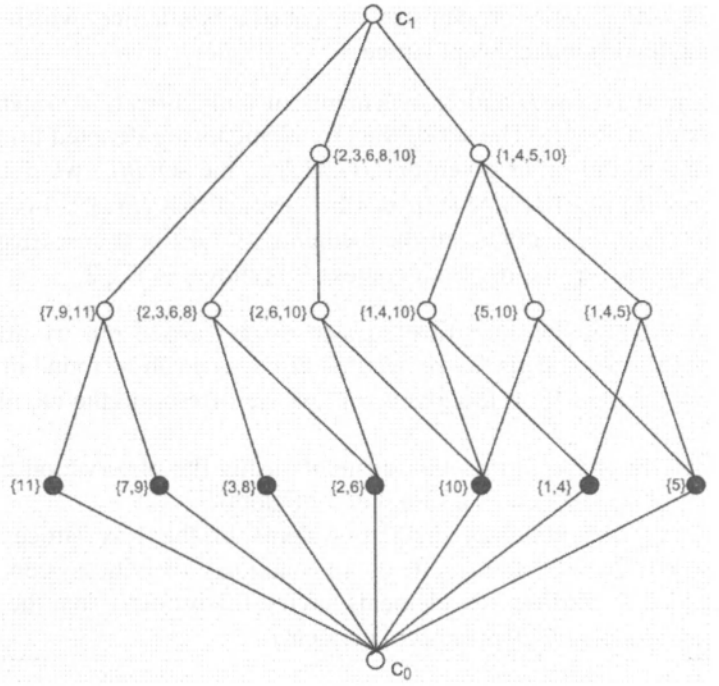

Figure 2. The box lattice of the context *K*. Atoms are represented by black circles, the further box elements are white. This lattice is much simpler than the concept lattice

Using step c) all of the classification systems were generated. In order to avoid uninteresting partitions two restrictions are assumed: every machine cell has to consist of at least three elements and the maximum number of bottleneck machines (i.e. machines processing parts from more than one family) is one. With these constraints four classification systems are formed (Table 2).

Table 2. Machine cells with the box lattice method

Machine cells			Bottleneck machine
{7,9,11}	{2,3,6,8}	{1,4,5,10}	
{7,9,11}	{2,3,6,8,10}	{1,4,5}	
{7,9,11}	{2,3,6,8}	{1,4,5}	{10}
{7,9,11}	{2,3,6,8}	{1,4,10 }	{5}

The context in Table 1 is borrowed from Cheng's study [5], where 12 algorithms for forming machine groups were compared. At the end of the examination there were three methods left yielding satisfactory results. These algorithms are the following: average linkage clustering (ALC) based on similarity coefficients [15], ZODIAC algorithm developed by Chandrasekharan and Rajagopalan [4] and a branch and bound algorithm (B&B) by Kusiak, Boe and Cheng [13]. The resulting machine cells are listed in Table 3.

Table 3. The results of three machine cell formation methods from Cheng's comparative study

Method	Machine cells			Bottleneck machines	
ALC	{7,9,11}	{2,3,6,8}	{1,4,5,10		
ZODIAC	{7,8,9,11	{2,3,6}.	{1,4,5,10		
B&B	{7,9,11}	{2,3,6}	{1,4,5}	{8}	{10}

It can be seen that these results are very similar to the decompositions obtained by the box lattice method, and our approach offers several opportunities to form the machine cells. The next step in improving the box lattice method is to solve the problem of constraints, namely how to build reasonable restrictions into the algorithm in order not to determine uninteresting classification systems.

Acknowledgements

This research was supported by the Hungarian National Foundation for Scientific Research (Grant No. T046913).

REFERENCES

[1] ADENSO-DÍAZ, B., LOZANO, S. EGUÍA, I.: *Part-machine grouping using weighted similarity coefficients.* Computers & Industrial Engineering, 48, pp. 553-570, 2005.

[2] BOULIF, M., ATIF, K.: *A new branch-&-bound-enhanced algorithm for the manufacturing cell formation problem.* Computers & Operation Research, 33, pp. 2219-2245, 2006 (available online from March 2005).

[3] BROWN, E. C., SUMICHRAST, R, T.: *Evaluating performance advantages of grouping genetic algorithms.* Engineering Applications of Artificial Intelligence, 18, pp. 1-12, 2005.

[4] CHANDRASEKHARAN, M. P., RAJAGOPALAN, R.: *ZODIAC-An algorithm for concurrent formation of part-families and machine-cells.* International Journal of Production Research, 25(6), pp. 835-850, 1987.

[5] CHENG, C. H.: *Algorithms for Grouping Machine Groups in Group Technology.* Omega, 20(4), pp. 493–501, 1992.

[6] DE LIT, P., FALKENAUER, E., DELCHAMBRE, A.: *Grouping genetic algorithms: an efficient method to solve the cell formation problem.* Mathematics and Computers in Simulation, 51, pp. 257-271, 2000.

[7] FALKENAUER, E.: *The grouping genetic algorithm-widening the scope of the GAs.* JORBEL-Belgian Journal of Operations Research, Statistics and Computer Science, 33, pp. 79-102, 1992.

[8] GANTER, B., WILLE, R.: *Formal Concept Analysis, Mathematical Foundations*, Springer Verlag, 1999.

[9] HOLLAND, J. H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control and Artificial Intelligence.* Ann Arbor, MI: University of Michigan Press, 1975.

[10] JEON, G., LEEP, H. R.: *Forming part families by using genetic algorithm and designing machine cells under demand changes.* Computers & Operations Research, 33, pp. 263-283, 2006 (available online from April 2005).

[11] KÖREI, A.: *Solving cell formation problems with concept lattices.* Machine Engineering, 5(3-4), pp. 37-46, 2005.

[12] KUSIAK, A.: *The Generalized Group Technology Concept.* International Journal of Production Research, 25(4), pp. 561-569, 1987.

[13] KUSIAK, A., BOE J. W., CHENG, C. H.: *Designing cellular manufacturing systems. Branch-and-bound and A* approaches.* IIE Transactions, 25(4), pp. 46-56, 1993.

[14] RADELECZKI, S.: *Classification systems and their lattice*, Discussiones Mathematicae General Algebra and Applications, 22, pp. 167-181, 2002.

[15] SEIFODDINI, H., WOLFE, P. M.: *Application of the similarity cofficient method in group technology.* IIE Transactions, 18(3), pp. 271-277, 1986.

[16] SHAFER, S. M.: *Part-Machine-Labour Grouping: The Problem and Solution Methods.* In: Group Technology and Cellular Manufacturing, State-of-the-Art Synthesis of Research and Practice, Edited by Suresh N.C. Kay J.M., Kluwer Academic Publishers, 1998.

[17] TÓTH, T., MOLNÁR, M.: *Computarized Similarity of Parts Supporting Group Technology.* Production Systems and Information Engineering, 1, pp. 71-82, 2003.

[18] WANG, J.: *Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm.* Automatica, 39, pp. 1607-1615, 2003.

[19] WEI, J. C., KERN, G. M.: *Commonality analysis: A linear cell clustering algorithm for group technology.* International Journal of Production Research, 27(12), pp. 2053-2062, 1989.

[20] WON, Y., LEE, K. C.: *Modified p-median approach for efficient GT cell formation.* Computers & Industrial Engineering, 46, pp. 495-510, 2004.

[21] YIN, Y., YASUDA, K.: *Similarity coefficient methods applied to the cell formation problem: A taxonomy and review.* International Journal of Production Economics, 101, pp. 329-352, 2006 (available online from March 2005).