

---

## A new mathematical approach to supporting group technology

---

Tibor Tóth

Department of Information Engineering,  
University of Miskolc,  
3515, Hungary  
E-mail: dr.tibor.toth@gmail.com

Sándor Radeleczki and Laura Veres

Department of Analysis,  
University of Miskolc,  
3515, Hungary  
E-mail: matradi@uni-miskolc.hu  
E-mail: matvlaura@gmail.com

Attila Körei\*

Department of Applied Mathematics,  
University of Miskolc,  
3515, Hungary  
E-mail: matka@uni-miskolc.hu  
\*Corresponding author

**Abstract:** This paper presents a mathematical approach which can be successfully applied in different group technology problems. The method is originated from formal concept analysis (FCA). We produce some particular extent partitions of a formal context which encodes the relationship between given technical objects and their attributes. We show how to define the part type notions and also how to solve the cell formation problem by means of these extent partitions. The main focus of the paper is a solution method for the determination of master parts. The method is based on notions defined by modal operators analogously to formal concepts and uses the extent partitions of the complementary context. It follows from the nature of the problem that several solutions can exist; a consistency measure is introduced in order to rank them. [Received 1 June 2012; Revised 12 April 2013; Accepted 17 June 2013]

**Keywords:** group technology applications; master parts; cell formation; formal concept analysis; FCA; extent partitions; modal operators; consistency measure.

**Reference** to this paper should be made as follows: Tóth, T., Radeleczki, S., Veres, L. and Körei, A. (xxxx) 'A new mathematical approach to supporting group technology', *European J. Industrial Engineering*, Vol. x, No. x, pp.xxx-xxx.

**Biographical notes:** Tibor Tóth received his PhD in Manufacturing Science and Technology in 1979 from the Hungarian Academy of Sciences (HAS). He received his DSc (Doctor of Science) degree in Engineering Sciences also from HAS in 1989. Since 1990, he has been a Professor at the University of Miskolc (UM). From 1989 to 2007, he was the Head of the Institute of Information Science at UM. His research fields are computer aided process planning (CAPP), production planning and control (PPC) and IT for manufacturing. He has significant research experience and practice in development of CAPP systems for industrial applications. He has published more than a hundred scientific works and his number of citations is over 400. Since 2001, he has been the Head of the Joseph Hatvany Doctoral School for Information Science, Engineering and Technology at UM.

Sándor Radeleczki received his PhD in Mathematics in 1996 from the Hungarian Academy of Sciences. He received his Habilitation in Information Science in 2009 from the University of Miskolc, Hungary. Currently, he is an Associate Professor in the Department of Analysis at the University of Miskolc. His research fields are lattice theory, partially ordered sets and their applications.

Laura Veres received her PhD in Information Science and Technology in 2010 at the University of Miskolc, Hungary. Currently, she is an Assistant Professor in the Department of Analysis at the University of Miskolc. Her research activity focuses on formal concept analysis especially on classification trees and rough sets, and its applications in group technology.

Attila Körei received his PhD in Information Science and Technology in 2008 from the University of Miskolc, Hungary. Currently, he is an Associate Professor in the Department of Applied Mathematics at the University of Miskolc. His research activity focuses on mainly formal concept analysis and its applications, and he also deals with problems from the area of numerical methods and genetic algorithms.

## 1 Introduction

At present there are numerous definitions for group technology (GT): it is described as a concept, philosophy, organising principle, methodology, to name a few. Of course, none of these can be considered as a complete definition. Each of them emphasises an aspect of GT, and moreover group manufacturing – which is another aspect – is not mentioned. It is expedient to consider GT as a synthesis of all the mentioned approaches. A generalised definition, for example, can be as follows: GT is the philosophy and methodology of recognising similarities among entities (problems, parts, machines, designs, technology processes, etc.) and utilising this knowledge during production.

One of the first proposals for defining the concept of GT was made by Mitrofanov in the 1950s (Mitrofanov, 1966; Niebel, 1965). As regards the fundamentals of GT, the works of Burbidge (1975) and Ham et al. (1985) are recommended. The conventional interpretation of those times can be considered as one of the most important preliminaries to computer aided process planning (CAPP) (Alting and Zhang,

1989; Studel, 1984). Today GT is considered as a basic principle that helps form a theoretical foundation for both CAPP and computer aided manufacturing (CAM) and that bridges the distance that would otherwise separate the two activities. GT is also a fundamental principle that is capable of providing a possibility for unifying and computer supporting all the functions of manufacturing; i.e., it can play an integrating role (Marri et al., 1998). From the point of view of mathematical formulations, the classical models proposed by Kusiak (1987) are especially influential.

Before the early eighties GT was a fairly practical approach (Houtzeel, 2001). In the last three decades, computer science and software technology have increased the efficiency of GT methodology to a great extent. We would like to underline the new possibilities of information technology (IT) in the field of GT in general. Nowadays the use of computer applications has already become everyday practice even in small and medium size enterprises (SMEs) (Mleczko, 2011). More and more often, SMEs are using software to increase the functionality of their products and offerings. In most cases computer aided design (CAD) and enterprise resource planning (ERP) application packages are used, which involve creating a data-base concerning the entities of the products to be manufactured. Recently product lifecycle management (PLM), as well as the use of data centre and cloud technology have increased the quality of services of the applications as well as reducing their costs. Data exchange between the applications is rapid and compatible. The speed of the searching procedures based upon similarity relations has increased considerably.

It seems to be trivial that repeated and systematic solving of any GT task will only be efficient in practice if data preparation and utilisation of results are also supported by computer. Hence, e.g.,  $(CAD + PLM) \rightarrow (GT + CAPP \text{ or/and } NCP) \rightarrow (MES \text{ or } SFC)$ , etc.; NCP – numerical control programming; MES – manufacturing execution system; SFC – shop floor control.

These factors greatly facilitate the application of GT methods and the principles of grouping, building and decomposing product entity sets can be extended to the manufacturing process as a whole. In the fields of product design, process planning and production planning there are numerous technical and economical tasks that can be solved by means of product data-bases in a more rapid and effective way, while additional cost of digitalisation occurs to a lesser and lesser degree. Such tasks include, for example, cell formation problems and the classic tasks of process planning, as well as tool set design, maintenance planning, supply chain logistics, storage capacity planning, etc. In CAPP, advantages of the application of GT methods have been well-known for a long time (Xu et al., 2011). Generating the master part models suitable for representing groups consisting of similar parts is greatly accelerated. Decomposition of part families (e.g., class, subclass, group, type) by computer applications can be carried out in a rapid and effective way.

## **2 Literature review**

At present the applicability of GT principles extends to an increasing number of synthesis-type tasks. Important research areas are as follows:

## 1 Theoretical foundation

Some characteristic objectives from the forefront of the fundamental research are as follows:

- a to elaborate appropriate abstract models
- b to carry out the technical modelling required
- c to create efficient solving algorithms according to the nature of the prevailing problem to be solved and taking into consideration all the constraints, as well as the objective function reasonably selected
- d to find adequate performance indices for qualifying the models, methods and algorithms respectively.

We mention several areas essential from the point of view of theoretical foundation: ontology, theory of production information engineering (IT for manufacturing included), theory of optimisation (with special regard to discrete manufacturing processes), and fundamentals of production logistics.

## 2 Abstract models

Among the models of this type the mathematical models and the combined abstract models (e.g., structural analysis and optimisation models, fuzzy and neuro-fuzzy models, different simulation models) can be taken into account. Let us consider, for example, the role of structural analysis. In this case it is expedient to distinguish three main additional viewpoints, namely:

- 1 verification of structural integrity
- 2 analysis of operational environment
- 3 modelling the manufacturing process.

As regards the approach of structural analysis, the robust numerical methods, high performance computing, as well as appropriate optimisation techniques are taken into consideration. Simulation models are very useful to transform the real world into treatable abstract models. However, there are some problems with simulation models, e.g., simulation can be coded in many different programming languages; expense of numerical techniques can be very large; analysis simulations frequently can not include optimisation techniques and in fairly frequent cases, because of the empiric character of the problem in question so-called metaheuristics have to be used (c.f. 4.). From the point of view of our paper the abstract models based upon formal concept analysis (FCA) are especially important.

The connection of ontology and FCA is shown in a comprehensive paper of Nanda et al. (2006), in which the authors introduce a novel methodology named product family ontology development methodology (PFODM) suitable for developing formal product ontologies by means of semantic web paradigm. In PFODM, FCA is used to identify similarities among finite sets of design artifacts

based on their properties. In the paper Web Ontology Language (OWL) is utilised to develop and refine a product family ontology.

### 3 Technical modelling

In this broad research field, manufacturing cell formation (or machine-part grouping) is especially important. The basic task in cellular manufacturing is how to group the machines into machine cells and the similar parts into part families. The comprehensive work of Chattopadhyay et al. (2012) proposes a visual clustering approach for machine-part cell formation using a self organising map (SOM) algorithm. This is an unsupervised neural network to achieve better GT efficiency measures of cell formation, as well as measures of SOM quality. This work also has established the criteria of choosing an optimum SOM map size based on the results of quantisation error, topography error, and average distortion measure during SOM training which have generated the best clustering and preservation of topology. To evaluate the performance of the proposed algorithm, the authors tested the several benchmark problems available in the literature. The results show that the proposed method not only generates the best and accurate solution as any of the results reported in literature so far, but also, in some instances the results produced are even better than the reported results earlier. It is considerable that the effectiveness of the proposed approach is also verified in statistical manner. Other publications will also be cited in the related section of our paper.

Another common research direction in technical modelling is process planning analysis for discrete manufacturing. Here it is expedient to distinguish the following frequented partial modelling and planning approaches (Xu et al., 2011): variant process planning, generative process planning, semi-generative process planning, knowledge-based planning (expert systems included), Feature recognition-based planning (based on CAD features), and STEP-based process planning (STEP is the standard for the exchange of product model data). Of course, the CAPP software systems in the market use partial models and methods from almost all of the theoretical backgrounds to a smaller or greater extent, in order to balance their benefits and shortcomings. There can be GT principles and partial models in many software systems because of the advantageous composition of clustering and objects. Flexible product design can also be supported by IT, because computer graphics-based attribute classification technology coupled with relational data-base systems can be used instead of cumbersome manual classification systems.

### 4 Solving algorithms formulated for models (metaheuristics)

As solution methods, among others, genetic algorithms (GA), artificial neural networks (ANN), simulated annealing (SA), tabu search (TS), particle swarm optimisation (PSO) and ant colony systems (ACS) can be taken into account (Megala et al., 2008; Ying et al., 2011). These are especially suitable for solving cell formation problem on the basis of machine-part incidence matrix. In the case of publications on metaheuristics, in general emphasis is on the algorithm itself. For instance, a detailed GA algorithm is proposed in the paper of Jahanzaib et al.

(2013). Recently another evolutionary algorithm, a bacteria foraging algorithm (BFA), has been proposed to solve cell formation considering operation time (Nouri and Hong, 2012). The new algorithm has been successfully tested with benchmark problems from the literature. Be careful: ‘complexity’ of several different algorithms (procedures) can only be compared on the basis of ‘pure computing need’ in that case if the stop of procedures (exit) is merely depending on input parameters and there is no other ‘degree of goodness’ (performance index, c.f.5.) In this case there is a possibility to carry out a comparison of the order of magnitude in an exact way. However, if accuracy, reliability, store reservation or any other type of performance index of the procedures in question is also significant then evaluation and assessment of the replies are competent on the basis of benchmark datasets accepted by the experts of the special field. Nevertheless, the charge of subjectivity can still occur here.

##### 5 Performance measures and computing demand (complexity) measures

Nowadays there are numerous tasks in the field of manufacturing engineering, which can be solved by means of the CAD-PLM database of the firms used increasingly, in a more rapid and efficient way. As an example, one can consider forming the manufacturing cells on the basis of parts’ similarity and prevailing technological environment and circumstances, as well as the production management; especially among the suppliers concerned. Theory and practice of the traditional cell forming mainly uses the so-called ‘group efficiency’ performance measures for qualifying the GT-based clustering algorithms. The performance indices proposed for measuring the performance of clustering procedures are well-known in the literature, such as *number of exceptional elements (EE)*, *grouping efficiency (GE)*, and *grouping efficacy (GC)* (Car and Mikac, 2006; Yina and Yasuda, 2005). Computer integrated systems, in addition to parts databases, can also use CAPP applications, which make it possible to create a database for *process plans* and *routings*. In this case there are facilities to utilise information on operation times and ‘set up’ times. To assess the effectiveness of the cells, then the *key performance indices* (KPI indices) offer more detailed and more precise possibility.

In the beginning the KPI indices were used for supporting the higher level (ERP) decision making of operation management (Kaplan and Norton, 2008), but in the course of the last ten years they have played more and more important role both at the level of production planning (MRP II) and at the MES level as well (Folan and Browne, 2005). Publication and introduction of the ISA-95 standard have strongly supported spreading the KPI indices (Scholten, 2007). At the higher levels of operation management, the physical output of production (product volume), the production costs, the utilisation of enterprise level capacities and means, the size of added value and the net profit have played the leading role.

In the measure of the performance of manufacturing systems, the following three type of indices have key-role: *utilisation rate*; *work in process* (or *stock level*) and *delivery capability* (*readiness for delivery*). The indices connecting with lead time and keeping the dead-line are close to these. Considering the manufacturing cells, other KPI indices can be very important, such as ‘interim material stream’,

variability of set up times, the cluster indices for tool sets, etc. (Busi, 2005). The modern cell formation methods, such as FCA make it possible a fine evaluation of the cells on the basis of KPI indices, taking into consideration the criterion named 'group efficiency'. This opportunity is very important for the suppliers, whose number is increasing in the manufacturing and engineering industries (Powell and Netland, 2010). The 'reconfigurable' flexible manufacturing can be interpreted as an extension of cellular manufacturing (Koren and Shpitalni, 2010). The objective of this new research area is to stop the contradiction between the longer recovery time of the FMS systems of high value and the much shorter term changes of the parts to be manufactured. The clustering methods based on up-to-date computerised database offer new possibilities in this area, too. A cluster analysis completed on the basis of different KPI indices can provide important input data for any reconfiguration or cell formation task (Shang and Tadikamalla, 1998).

Scientific investigations have been mainly carried out in these directions.

It is easy to see that, in order to realise the advantages of GT, there are many promising models and methods in this broad-based philosophy of manufacturing. However, the techniques for the different fields – for part-machine grouping, engineering databases, expert system-based design methods for identifying part families, new analytical and simulation tools for evaluating performance of cells, etc., depend on the prevailing IT tools. Therefore it is expedient to search for the new mathematical models and algorithms which are more suitable for the new IT tools.

In this paper the focus is on generating the master part of a part group (see Section 5). Our method is based on extent partitions and some specially defined modal operators. The elaborated FCA-based approach is also applied to solve the cell formation problem. For the cell formation problem we compare our method to other methods in the literature. As regards the FCA-based master part generation problem, we did not find any previous attempts in the literature.

### 3 Elements of FCA

FCA is a field of applied lattice theory that mainly deals with conceptual hierarchy. In the last 20 years FCA has proved to be a useful tool in different scientific fields, e.g., in knowledge representation, data mining, natural language processing, and architecture (Ganter et al., 2005). Regarding GT, a very practical application of FCA was introduced in Nanda et al. (2007), where FCA tools provide a structured framework of information management in product family designing processes. Another application in GT was presented in Belohlavek et al. (2009), where formal concepts were used in solving the cell formation problem. The problem and our new, FCA-based solution method are discussed in Section 4.3.

This section provides a brief introduction to FCA based on the fundamental work of Ganter and Wille (1999). The main tools of FCA are concept lattices which can be originated from formal contexts. A *formal context*  $\mathbb{K} = (G, M, I)$  consists of two sets  $G$  and  $M$  and a binary relation  $I \subseteq G \times M$  between the elements of  $G$  and  $M$ . The elements of the set  $G$  are called *objects* and the elements of  $M$  are called *attributes* of the context; in order to express that the object  $g \in G$  has the attribute  $m \in M$  we write  $gIm$  or  $(g, m) \in I$ .

A finite context can be easily represented by a *binary* or a *cross table*. This is a two-dimensional table whose rows are headed by the object names and columns by the attribute names. A digit 1 (or a cross) in row  $g$  and column  $m$  means that the object  $g$  has the attribute  $m$ , i.e.,  $(g, m) \in I$ , and we insert a zero in the cell (or leave it void) if  $(g, m) \notin I$ . Table 1 presents a binary table containing seven objects and six attributes.

**Table 1** Example of a formal context

|       | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | $m_6$ |
|-------|-------|-------|-------|-------|-------|-------|
| $g_1$ | 0     | 1     | 0     | 0     | 0     | 1     |
| $g_2$ | 1     | 0     | 0     | 0     | 0     | 0     |
| $g_3$ | 0     | 1     | 0     | 0     | 1     | 1     |
| $g_4$ | 0     | 1     | 1     | 1     | 1     | 0     |
| $g_5$ | 0     | 1     | 1     | 0     | 0     | 1     |
| $g_6$ | 1     | 1     | 0     | 1     | 0     | 0     |
| $g_7$ | 0     | 1     | 1     | 0     | 1     | 1     |

Starting from a formal context  $(G, M, I)$  we can define several mappings between the subsets of  $G$  and  $M$ . We will give the mathematical formulation of some GT terms with the aid of these operations.

For a set  $A \subseteq G$  of objects we define

$$A^I := \{m \in M \mid gIm \text{ for all } g \in A\} \quad (1-a)$$

$$A^\diamond := \{m \in M \mid gIm \text{ for at least one } g \in A\} \quad (1-b)$$

$$A^\square := \{m \in M \mid gIm \Rightarrow g \in A\}. \quad (1-c)$$

It is easy to see that  $A^I$  is the set of attributes common to all objects of  $A$ ,  $A^\diamond$  is the set of all attributes which appear for at least one object belonging to  $A$ , and  $A^\square$  is the set of those attributes that can appear only for some object from  $A$ . For example, considering the context in Table 1 and the set  $A = \{g_4, g_5, g_7\}$  we obtain that  $A^I = \{m_2, m_3\}$ ,  $A^\diamond = \{m_2, m_3, m_4, m_5, m_6\}$  and  $A^\square = \{m_3\}$ .

Similarly, for a set  $B \subseteq M$  of attributes we also can define

$$B^I := \{g \in G \mid gIm \text{ for all } m \in B\} \quad (2-a)$$

$$B^\diamond := \{g \in G \mid gIm \text{ for at least one } m \in B\} \quad (2-b)$$

$$B^\square := \{g \in G \mid gIm \Rightarrow m \in B\}. \quad (2-c)$$

Definitions (1-a) and (2-a) are the well-known operators from Ganter and Wille (1999); (1-b), (1-c), (2-b) and (2-c) define modal operators which were used in Yao and Chen (2006) for connecting FCA to rough sets theory.

A *formal concept* of the context  $(G, M, I)$  is a pair  $(A, B)$  with  $A \subseteq G$ ,  $B \subseteq M$ ,  $A^I = B$  and  $B^I = A$ .  $A$  is called the *extent* and  $B$  the *intent* of the concept  $(A, B)$ . For instance, see again the context in Table 1, where the sets  $A = \{g_4, g_5, g_7\}$  and  $B = \{m_2, m_3\}$  form a concept.

It is easy to check that for any subset  $A \subseteq G$ , the pair  $(A^{II}, A^I)$  is always a concept, moreover:

$$A \text{ is a concept extent} \Leftrightarrow A^{II} = A. \quad (3)$$

$\mathfrak{B}(G, M, I)$  denotes the set of all concepts of the context  $(G, M, I)$ , which can be ordered as follows:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1).$$

The relation  $\leq$  is called the *hierarchical order* of the concepts. The partially ordered set  $(\mathfrak{B}(G, M, I), \leq)$  constructed in this way is in fact a complete lattice: it is called the *concept lattice* of the context  $(G, M, I)$ . A concept lattice can be demonstrated by its Hasse-diagramme, which always has a bottom and a top element corresponding to the least and the greatest concept of the lattice. An example of a Hasse-diagramme – generated from the context of Table 1 – is presented in Figure 1. Every rectangle of the diagramme corresponds to a concept of the lattice, containing its extent (first row) and its intent (second row). The lines connecting the rectangles demonstrate the hierarchy of the concepts. The shadowed rectangles denote the concepts whose extents belong to the finest extent partition of the context (see Section 4.1).

It is important to mention an other construction which is similar to concept lattices. Let us denote by  $\mathcal{H}(G, M, I)$  the set of all those pairs  $(A, B)$  with  $A \subseteq G$ ,  $B \subseteq M$  that satisfy the relations  $B = A^\diamond$ ,  $A = B^\square$ . We will call them *analogue concepts*. It was shown by Yao and Chen (2006) that the set of analogue concepts can be partially ordered as follows:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_1 \subseteq B_2),$$

and the partially ordered set  $(\mathcal{H}(G, M, I), \leq)$  is a complete lattice again (see also Veres (2010)). Similarly to (3) it is easy to check that for any subset  $A \subseteq G$ ,  $(A^{\diamond\square}, A^\diamond) \in \mathcal{H}(G, M, I)$ .

#### 4 Extent partitions and their direct applications in GT

A *partition* of a set  $G$  is a family of nonempty, mutually disjoint subsets of  $G$  whose union is  $G$ , i.e., a family

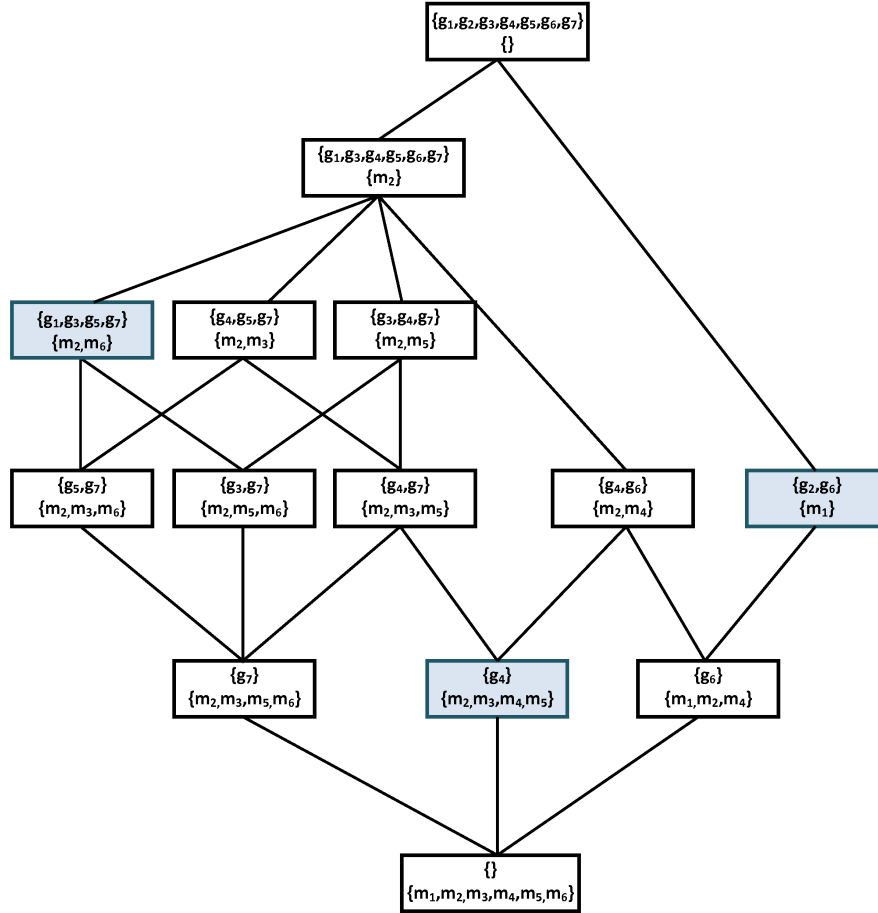
$$\pi = \{G_t \mid t \in T\}$$

of sets  $G_t \subseteq G$  (where  $T$  is some index set) with  $G_t \neq \emptyset$ ,  $G_s \cap G_t = \emptyset$  for all  $s \neq t \in T$  and  $\bigcup_{t \in T} G_t = G$ . The sets  $G_t$  are called the *classes* or *blocks* of the partition  $\pi$ .

We define an ordering relation between the partitions of a given set. For partitions  $\pi_1$  and  $\pi_2$  of  $G$  we write

$$\pi_1 \leq \pi_2$$

iff each class of  $\pi_1$  is contained in some class of  $\pi_2$ . We then say that  $\pi_1$  is *finer* or equal to  $\pi_2$ , or, equivalently, that  $\pi_2$  is *coarser* than  $\pi_1$ , and sometimes, that  $\pi_1$  is a *refinement* of  $\pi_2$ .

**Figure 1** Concept lattice generated from the context in Table 1 (see online version for colours)

#### 4.1 Extent partitions of a formal context

If the set  $G$  is an object set of a formal context  $(G, M, I)$  then special partitions of it can exist. Ganter et al. (2013) suggested the following definition:

*Definition 4.1:* An *extent partition* of the formal context  $(G, M, I)$  is a partition  $\pi = \{A_t \mid t \in T\}$  of the object set  $G$ , having the property  $A_t^I = A_t$  for all  $t \in T$ .

Note that since the intersection of extents always yields an extent, the common refinements of extent partitions are again extent partitions. In particular, there is always a *finest* (or *least*) *extent partition* of  $(G, M, I)$  which we denote by  $\pi^\circ$ . For example the least extent partition of the context presented in Table 1 is  $\pi^\circ = \{\{g_4\}, \{g_2, g_6\}, \{g_1, g_3, g_5, g_7\}\}$ . The concepts belonging to the blocks of  $\pi^\circ$  (i.e., the atomic box extents) can be identified as the shadowed rectangles in Figure 1.

The following algorithm is introduced in Körei and Radeleczki (2006) for determining the least extent partition  $\pi^\circ$  of an arbitrary context  $(G, M, I)$ .

---

**Algorithm 1**


---

```

begin
   $\pi^\circ := \{g^{II} \mid g \in G\};$ 
  while there are  $X, Y \in \pi^\circ, X \neq Y$  with  $X \cap Y \neq \emptyset$ 
  do  $\pi^\circ := (\pi^\circ \setminus \{X, Y\}) \cup \{(X \cup Y)^{II}\}$ 
end

```

---

The computational complexity of the algorithm is  $O(n^2k)$  where  $n$  is the number of the objects and  $k$  is the number of the attributes.

#### 4.2 Part types

In what follows we will consider a formal context  $\mathbb{K} = (G, M, I)$  with a concrete and fixed technical meaning, where  $G$  denotes a fixed (finite, nonempty) set of (technical) objects,  $M$  denotes a fixed (finite, nonempty) set of some possible, technically relevant properties, and for any  $g \in G$  and any  $m \in M$   $gIm$  or  $(g, m) \in I$  will mean that ‘The object (part)  $g$  has the property  $m$ ’. Additionally we suppose that the context does not contain rows or columns filled with only zeros. A full zero column means that the corresponding property not held by any of the parts, so it is irrelevant; a full zero row corresponds to a part  $g$  possessing none of the properties from our list. The presence of such an object would imply that the only possible extent partition is the trivial  $\pi = \{G\}$ . Indeed, in this case  $\{g\}^I = \emptyset$  and  $\{g\}^{II} = \emptyset^I = G$ , hence for the finest extent partition we have  $\pi^\circ = \{G\}$ .

According to Mitrofanov’s (1966) definition, partitioning the objects set into part types means that we are looking for the finest partition  $\{A_t \mid t \in T\}$  of the set  $G$  having the property that its classes are characterised by the common attributes of the objects (parts) belonging to this class. More explicitly, this definition of Mitrofanov means that any object  $g$  that shares all the common properties of the objects from the class  $A_t$  (that is, all attributes  $m \in A_t^I$ ) must belong also to the class  $A_t$ . In mathematical terms this means that the relation

$$A_t^{II} = A_t$$

must be satisfied for all  $t \in T$ . According to (3) this equation yields that the classes  $A_t$  are concept extents for all  $t \in T$  and they form the finest extent partition  $\pi^\circ$  of the context  $(G, M, I)$ .

The algorithm presented in Section 4.1 provides an exact method for determining the part types, and its running time is a cubic of the input parameters.

#### 4.3 Machine cells

One of the main tasks in cellular manufacturing is the formation of machine cells and their corresponding product families. This task is often referred to as the cell formation problem or machine-part grouping problem, and a large number of different methods

have been suggested to solve it. Regarding our topic the most relevant solution method is introduced in Belohlavek et al. (2009), because the algorithms elaborated by the authors use formal concepts in the process of forming machine cells. We show that extent partitions are also suitable tools for solving the cell formation problem and that our solution is similar in quality to those obtained by other techniques.

In most cases the cell formation problem is formulated by a machine-part incidence matrix  $A = [a_{ij}]$ , where

$$a_{ij} = \begin{cases} 1, & \text{if machine } i \text{ processes part } j, \\ 0, & \text{otherwise.} \end{cases}$$

The aim is to rearrange the rows and columns of the incidence matrix in order to obtain a block diagonal matrix, from which the groups of machines and the corresponding part families can be identified. In an ideal case there are no ones outside the diagonal blocks (there are no exceptional parts) and no zeros inside the blocks (there are no voids). In general the objective of the grouping process is to maximise the intra-cell machine utilisation (the fewer voids the better) and to minimise the inter-cell movement of the parts (the fewer exceptional elements the better). To avoid the subjective judgement of goodness of admissible solutions, a number of measures are suggested in the literature for evaluating them. The most frequently used indicator is *grouping efficacy*, introduced by Kumar and Chandrasekharan (1990) as follows:

$$\Gamma = \frac{n_1 - n_e}{n_1 + n_0},$$

where  $n_1$  is the number of the operations (the number of the ones in the incidence matrix),  $n_e$  is the number of the exceptional parts and  $n_0$  is the number of the voids.

A given machine-part incidence matrix can be considered as a formal context  $(G, M, I)$ , where  $G$  is the set of the machines,  $M$  contains the parts and  $I$  is determined by the incidence matrix  $A$  in the natural way, namely if  $g$  is the  $i^{\text{th}}$  machine and  $m$  is the  $j^{\text{th}}$  part then  $(g, m) \in I \Leftrightarrow a_{ij} = 1$ . Based on the extent partitions of  $(G, M, I)$  the machine-part grouping problem can be solved, because the machine cells can be identified as the blocks of an extent partition of  $G$ . A concept lattice can induce numerous extent partitions, but it is not necessary to generate all of them to solve the problem. In most cases the blocks of the suitable partitions are maximal in the following sense: the supremum of the concepts induced by any two blocks is equal to the maximal element of the concept lattice. In other words, if  $\{B_k \mid k \in K\}$  is a partition of  $G$ ,  $B_i^I \cap B_j^I = \emptyset$  must be held for every  $i \neq j$ . This condition assures that there are no parts which are to be processed by all of the machines belonging to different machine cells.

After selecting the maximal extent partitions we have to determine the associated part families. Obviously, if  $B_i$  is a machine cell, the elements of the set  $B_i^I$  are to be processed by all machines in  $B_i$ , so  $B_i^I \subset P_i$ , where  $P_i$  denotes the part family belonging to the machine cell  $B_i$ . There can be parts, however, which are not counted into any family in this way. For such a part  $p$  we calculate a quotient for every machine cell as follows:

$$q_k(p) = \frac{\text{the number of the machines in } B_k \text{ processing } p}{\text{the number of the machines in } B_k}$$

and  $p$  will be inserted into that family  $P_k$  for which  $k$  the value  $q_k(p)$  is maximal. The aim of this method is to reach the highest grouping efficacy possible when forming part families for the blocks of a given extent partition. Finally, if we have more than one solution for the problem, we have to choose the extent partition (and machine cells at the same time) for which the grouping efficacy is maximal. A more detailed discussion of the calculation is provided in Tóth and Körei (2006).

**Table 2** Comparison of the solutions obtained by extent partitions with the results of other methods

| <i>Problem</i>       | <i>1</i>            |                   | <i>2</i>             |                           |
|----------------------|---------------------|-------------------|----------------------|---------------------------|
| Problem size         | 12 × 10             |                   | 15 × 10              |                           |
| Solution method      | Integer programming | Extent partitions | Clustering algorithm | Extent partitions         |
| Number of cells      | 3                   | 3                 | 3                    | 3                         |
| Cell utilisations    | 0.81; 0.87; 0.89    | 0.83; 0.83; 0.89; | 0.81; 0.93; 1        | 0.85; 0.93; 1             |
| Voids                | 6                   | 6                 | 4                    | 4                         |
| Exceptional elements | 5                   | 6                 | 0                    | 0                         |
| <i>Problem</i>       | <i>3</i>            |                   | <i>4</i>             |                           |
| Problem size         | 8 × 20              |                   | 14 × 24              |                           |
| Solution method      | Similarity matrix   | Extent partitions | Similarity matrix    | Extent partitions         |
| Number of cells      | 3                   | 3                 | 4                    | 5                         |
| Cell utilisations    | 1; 1; 1             | 1; 1; 1           | 0.5; 0.63; 0.64; 1   | 0.56; 0.69; 0.92; 0.79; 1 |
| Voids                | 0                   | 0                 | 29                   | 18                        |
| Exceptional elements | 9                   | 9                 | 4                    | 9                         |

In order to compare our approach with other methods we analysed the same benchmark datasets which were used in Belohlavek et al. (2009). The four examples are from three different papers: Problem 1 is presented in Albadawi et al. (2005) and an integer programming method was suggested to solve it, Problem 2 is from Chan and Milner (1982) and solved by an iterative clustering algorithm and Problems 3 and 4 are discussed in Srinivasan et al. (1990) and similarity matrices are used to solve them. The problems are solved using the extent partitions method and all solutions are evaluated considering the total number of voids and exceptional elements and the cell utilisation ratios of the created machine cells. Table 2 contains the results of the comparison.

For the sake of simplicity Table 2 does not contain the results obtained by the proposed method in Belohlavek et al. (2009), because they calculated three different solutions for each dataset. Comparing our results with the solutions presented in Belohlavek et al. (2009) and the solutions obtained by the original approaches, we can say that our method performs well in terms of the selected parameters. According to the complexity of the used algorithms, in a particular case the determination of all

extent partitions may need exponential computing time (if every concept is a block of some extent partition). Because of the nature of the cell formation problem this opportunity can be excluded and the solutions can be calculated in a reasonable time. As a matter of fact, our algorithms are implemented in MATLAB programs and the results presented in Table 2 are obtained in a couple of seconds using a standard notebook computer.

## 5 Master parts

In a GT-based approach the parts that are similar in terms of their design or manufacturing operations are grouped into one family. For every part family we can determine a special, not necessarily existing object, the *master part* of the family, which possesses all the attributes of the parts belonging to the family. From a mathematical viewpoint master parts are used to form a partition of the set  $G$  of technical objects, namely we are looking for partitions  $\pi = \{A_j \subseteq G \mid j \in J\}$  whose classes  $A_j$  are characterised by the set of all those attributes, each of which appears for at least one object of this class, that is, by the sets  $A_j^\diamond$ . In mathematical formulation this means that  $g \in A_j \Leftrightarrow g \in A_j^{\diamond\Box}$ , i.e.,

$$A_j^{\diamond\Box} = A_j, \text{ for all } j \in J \quad (4)$$

Now, this partition must contain as small a number of classes as possible. This means that our partition  $\pi$  must be coarser than the extent partition  $\pi^\circ$ . Therefore, we are asking for partitions  $\pi = \{A_k \subseteq G \mid k \in K\}$  of  $G$  for which (4) holds for all  $k \in K$ , such that the number  $|K|$  of their blocks is minimal (but  $|K| \geq 2$ , if it is possible), and such that  $\pi^\circ \leq \pi$ . Clearly, the trivial partition  $\nabla$  containing the whole set  $G$  as a single block is always a solution of this problem, with  $|K| = 1$ , since  $G^{\diamond\Box} = G$ , and  $\nabla = \{G\}$  is always an extent partition, hence  $\pi^\circ \leq \nabla$ . Therefore we ask for solutions with a minimal number  $K$ , but  $\pi < \nabla$ . In the majority of real life cases there are several nontrivial solutions, such that their number of blocks is minimal (except  $\nabla$ ). Therefore, the user must select a solution based on the technical characteristics of the concrete task. Of course, even the structure of these blocks can indicate those partitions that are far from the practical solutions. In order to assist in selecting among the different solutions we introduce an indicator which is suitable for evaluating the partitions obtained by our approach. First we define a so-called consistency measure for qualifying the blocks, then by calculating the mean of these values we have a performance measure valid for a partition.

Let  $\pi = \{A_k \subseteq G \mid k \in K\}$  be a partition of the object set  $G$  and  $A_j$  be a block of this partition. The set  $A_j^\diamond$  contains all the attributes of the objects belonging to  $A_j$ , hence the direct product  $A_j \times A_j^\diamond$  can be visualised as a rectangle containing all the ones corresponding to the objects from  $A_j$ . Let  $X_j$  denote the number of the ones from this rectangle, and let  $Y_j$  be the number of the ones from the rectangle corresponding to  $\overline{A_j} \times A_j^\diamond$ , where  $\overline{A_j}$  stands for the complementary set of  $A_j$  (i.e.,  $\overline{A_j} := G \setminus A_j$ ). More precisely,  $X_j = \sum_{p \in A_j} |p'|$  and  $Y_j = \sum_{p \in \overline{A_j}} |p' \cap A_j^\diamond|$ . Normalising the numbers  $X_j$  and  $Y_j$  by the cardinality of the set  $A_j \times A_j^\diamond$ , we obtain ratios indicating to what extent

the attributes of the set  $A_j^\diamond$  characterise the object sets  $A_j$  and  $\overline{A_j}$  respectively. The difference of these ratios is defined as the *consistency measure* of the block  $A_j$ :

$$c(A_j) := \frac{X_j - Y_j}{|A_j \times A_j^\diamond|}. \quad (5)$$

The greater  $c(A_j)$  is, the more consistent the set  $A_j$  is, according to the set of the attributes of its objects. The *consistency*  $C(\pi)$  of a partition  $\pi = \{A_k \subseteq G \mid k \in K\}$  is defined as the arithmetical mean of the consistency measure of their blocks, that is:

$$C(\pi) := \frac{\sum_{k \in K} c(A_k)}{|K|}. \quad (6)$$

Finally, the master part belonging to a class  $A_k$  of  $\pi$  will be interpreted as a list of attributes which appear for at least one object  $g \in A_k$ , in other words, it will be identified with the set  $A_k^\diamond \subseteq M$ .

Definitions (1-a), (1-b), (1-c) and the definition of consistency measure imply the following lemma, which gives boundaries for the consistency of a partition.

*Lemma 5.1:* For any partition  $\pi = \{A_k \subseteq G \mid k \in K\}$  we have  $-1 < C(\pi) \leq 1$ , and  $C(\pi) = 1 \iff A_k^I = A_k^\diamond = A_k^\square$  for every  $k \in K$ .

As a consequence of this lemma it may be the case that there are partitions with negative consistency. This occurs when the consistency measure of the blocks of a partition are mainly negative, for example the blocks are small (having only a few elements), or they are not sufficiently consistent, i.e., the attributes of their objects can also be typical for the objects belonging to other blocks. In general these partitions are not suitable for determining the master parts, so we focus on the solutions with positive consistency.

### 5.1 Using complementary context for determining master parts

For a formal context  $\mathbb{K} = (G, M, I)$ , its complementary context is defined as the context  $\mathbb{K}^C = (G, M, I^C)$ , consisting of the same object set  $G$  and attribute set  $M$ , but with the property that the incidence relation  $I^C \subseteq G \times M$  is just the set theoretical complement of  $I$ , that is

$$I^C = (G \times M) \setminus I.$$

Hence  $I^C$  is satisfied for some pair  $g \in G$  and  $m \in M$ , if and only if the relation  $(g, m) \in I$  is not satisfied, that is the object  $g$  does not have the attribute  $m$ .

*Lemma 5.2:* For any set  $A \subseteq G$  we have  $A^{\diamond\square} = A^{I^C I^C}$ .

*Proof:* We have  $A^\diamond = \{m \in M \mid gIm \text{ for some } g \in A\} = M \setminus \{m \mid (g, m) \notin I, \text{ for all } g \in A\} = M \setminus \{m \mid (g, m) \in I^C, \text{ for all } g \in A\} = M \setminus A^{I^C}$  and hence by (2-c)  $A^{\diamond\square} = (M \setminus A^{I^C})^\square = \{g \in G \mid gIm \implies m \in M \setminus A^{I^C}\} = \{g \in G \mid (g, m) \notin I, \text{ for all } m \in A^{I^C}\} = \{g \in G \mid (g, m) \in I^C, \text{ for all } m \in A^{I^C}\} = A^{I^C I^C}$ , using (2-a) in the last equality.

As an immediate consequence of this lemma and Definition 4.1 we obtain

*Corollary 5.3: A partition  $\pi = \{A_j \mid j \in J\}$  of the object set  $G$  has the property that  $A_j^{\diamond\Box} = A_j$ , for all  $j \in J$  if and only if  $\pi$  is an extent partition of the complementary context  $\mathbb{K}^C = (G, M, I^C)$ .*

Using this Corollary the master parts and the partition  $\pi$  esponding to them can be obtained as follows:

- 1 Using Algorithm 1, the finest extent partition  $\pi^\circ = \{B_j \mid j \in J\}$  of the context  $\mathbb{K} = (G, M, I)$  is generated.
- 2 If  $\pi^\circ \neq \{G\}$  then we derive the complementary context  $\mathbb{K}^C = (G, M, I^C)$  from the context  $\mathbb{K} = (G, M, I)$ , where  $I^C = (G \times M) \setminus I$ .

Otherwise we obtain  $\pi = \{G\}$  and we are ready, since we have a single master part  $c(G)$  possessing all of the attributes from the set  $M$  (see Section 5.2, Example 2).

- 3 We form those maximal, nontrivial extent partitions  $\pi^* = \{A_k \mid k \in K\}$  of the context  $\mathbb{K}^C$  which satisfy the property  $\pi^* \geq \pi^\circ$ ,  $\pi^* \neq \nabla$ . Their nonempty set will be denoted by  $\mathcal{M}$ .
- 4 We compute the consistency measure  $C(\pi)$  of each partition  $\pi \in \mathcal{M}$ , and the solutions with positive measures are listed. Then the decision between these listed partitions is made by the analysis of their technical content.
- 5 When the most reasonable partition  $\pi = \{A_k \mid k \in K\}$  is chosen, using the original context  $\mathbb{K}$  we form the attribute sets  $A_k^\diamond$ ,  $k \in K$ . Then a master part  $c(A_k)$  is defined as a virtual technical object having as its list of attributes the set  $A_k^\diamond$  (see Section 5.2, Example 1).

We note that in Step 3 of the above method we use the same algorithm for selecting the maximal nontrivial extent partitions as in the case of the cell formation problem and in Step 4 the contribution of an engineering expert is supposed.

## 5.2 Examples

In the following we suppose that the feature-based representation of a part includes, at a higher abstraction level, the geometric forms with their sizes, tolerances and relations, as well as the manufacturing technology available.

In order to illustrate the determination process of the master parts we consider two examples. In both problems different parts are given and we use the following attributes for characterising them:

- a fictitious endface
- b open cylindrical mantle
- c semi-closed cylindrical mantle
- d external conical surface

- e external thread
- f external recessing slot
- g external bevel
- h hole
- i internal conical surface
- j internal thread
- k internal recessing slot
- l internal bevel
- m machinable in one clamping
- n multiple stepped outside
- o multiple stepped inside
- p hole without steps.

In *Example 1* ten parts are examined (Figure 2), and using the above attributes a  $10 \times 16$  type context can be generated (Table 3).

**Figure 2** Parts of Example 1

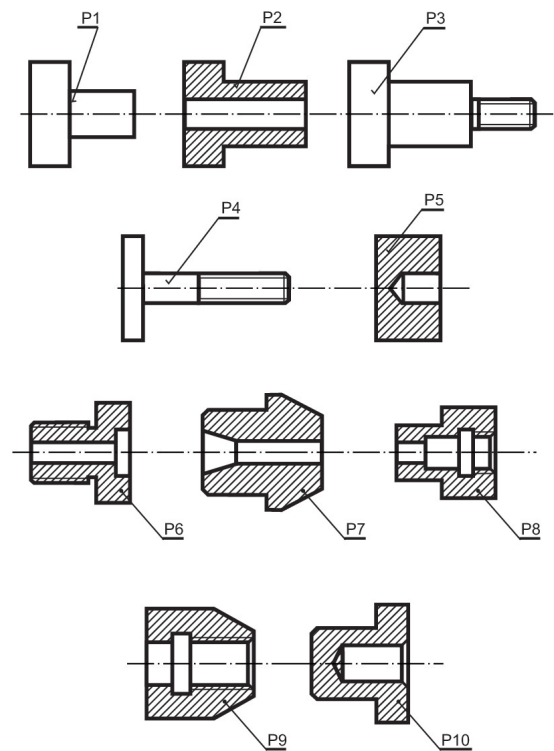


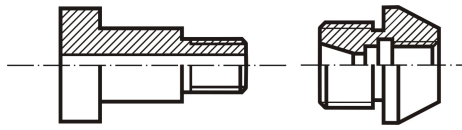
**Table 3** Formal context for Example 1

|     | <i>a</i> | <i>b</i> | <i>c</i> | <i>d</i> | <i>e</i> | <i>f</i> | <i>g</i> | <i>h</i> | <i>i</i> | <i>j</i> | <i>k</i> | <i>l</i> | <i>m</i> | <i>n</i> | <i>o</i> | <i>p</i> |
|-----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| P1  | 0        | 1        | 1        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 0        |
| P2  | 0        | 1        | 1        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 1        |
| P3  | 1        | 1        | 1        | 0        | 1        | 0        | 1        | 0        | 0        | 0        | 0        | 0        | 1        | 1        | 0        | 0        |
| P4  | 1        | 1        | 1        | 0        | 1        | 0        | 1        | 0        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 0        |
| P5  | 1        | 1        | 0        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 1        |
| P6  | 0        | 1        | 1        | 0        | 1        | 1        | 1        | 1        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |
| P7  | 1        | 1        | 1        | 1        | 0        | 0        | 1        | 1        | 1        | 0        | 0        | 0        | 0        | 0        | 0        | 1        |
| P8  | 0        | 1        | 1        | 0        | 0        | 0        | 0        | 1        | 0        | 1        | 1        | 1        | 0        | 0        | 1        | 0        |
| P9  | 1        | 1        | 0        | 1        | 0        | 0        | 0        | 1        | 0        | 1        | 1        | 1        | 0        | 0        | 0        | 0        |
| P10 | 1        | 1        | 1        | 0        | 0        | 0        | 1        | 1        | 0        | 0        | 0        | 1        | 0        | 0        | 0        | 1        |

After finding the partitions  $\pi = \{A_j \subseteq G \mid j \in J\}$  for which  $A_j^{\diamond\Box} = A_j$ , for all  $j \in J$ , we determined the different maximal nontrivial solutions. Then the consistency measure of these solutions were calculated, and we selected those for which positive values resulted. The remaining solutions are the following:

$$\begin{aligned}
\pi_1 &= \{(P1, P2, P3, P4, P5, P6); (P7, P8, P9, P10)\} \text{ with } C(\pi_1) = 0.1896, \\
\pi_2 &= \{(P1, P2, P3, P4, P5); (P6, P7, P8, P9, P10)\} \text{ with } C(\pi_2) = 0.1405, \\
\pi_3 &= \{(P1, P2, P3, P4, P5, P7); (P6, P8, P9, P10)\} \text{ with } C(\pi_3) = 0.1212, \\
\pi_4 &= \{(P1, P2, P3, P4, P5, P6, P7); (P8, P9, P10)\} \text{ with } C(\pi_4) = 0.0395, \\
\pi_5 &= \{(P1, P2, P3, P4, P5, P7, P10); (P6, P8, P9)\} \text{ with } C(\pi_5) = 0.0218.
\end{aligned}$$

The user is recommended to choose one of these solutions, taking into careful consideration the technical parameters of the required task. Figure 3 demonstrates the master parts belonging to the solution  $\pi_2$ .

**Figure 3** Master parts for Example 1

Example 2 includes eight parts (Figure 4) and the context generated for them is given in Table 4. Starting from this context we only obtain the trivial partition  $\nabla = \{G\}$ , the master part belonging to this solution is shown in Figure 5.

## 6 Conclusions

In this paper we intended to show how FCA methods can be applied to supporting GT. It is important to remark that we cannot expect an exact solution in the case of every application using our methods. We have obtained an exact solution for determination of part types; they are determined in their entire extent from a mathematical point of

view by their attributes. In the case of applications concerning machine cells and master parts, it can occur, as a consequence of both the engineering and the mathematical nature of the tasks, that several solutions exist that meet some kind of criteria. In this case the optimum solution from a technological point of view has to be selected by the production engineer. For instance, when determining master parts we often find several maximal extent partitions; their consistency measure may be only one of the significant indicators, therefore a well-founded knowledge of the engineering background is required to select the most suitable partition. Thus, our proposed solutions for machine cells and master parts are necessarily of a heuristic nature.

**Figure 4** Parts of Example 2

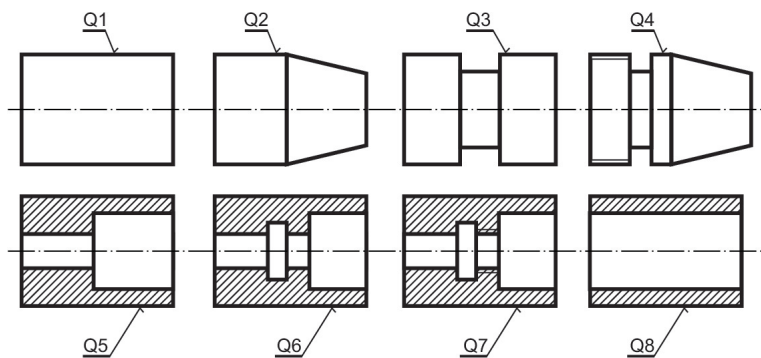
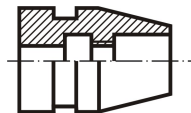


**Table 4** Formal context for Example 2

|    | <i>a</i> | <i>b</i> | <i>d</i> | <i>e</i> | <i>f</i> | <i>h</i> | <i>j</i> | <i>k</i> | <i>p</i> |
|----|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Q1 | 0        | 1        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |
| Q2 | 1        | 1        | 1        | 0        | 0        | 0        | 0        | 0        | 0        |
| Q3 | 0        | 1        | 0        | 0        | 1        | 0        | 0        | 0        | 0        |
| Q4 | 0        | 1        | 1        | 1        | 1        | 0        | 0        | 0        | 0        |
| Q5 | 0        | 1        | 0        | 0        | 0        | 1        | 0        | 0        | 0        |
| Q6 | 0        | 1        | 0        | 0        | 0        | 1        | 0        | 1        | 0        |
| Q7 | 0        | 1        | 0        | 0        | 0        | 1        | 1        | 1        | 0        |
| Q8 | 0        | 1        | 0        | 0        | 0        | 1        | 0        | 0        | 1        |

**Figure 5** Master part for Example 2



At last, we would like to emphasise that FCA is not only a newer mathematical tool to complete the list of metaheuristics suitable for supporting GT and available at present. We think, because of the special properties of the method, FCA also has opened manifold generalisation possibilities in different directions of application.

## Acknowledgements

The research work was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-001 project with support by the European Union, co-financed by the European Social Fund. The authors also sincerely acknowledge the helpful criticism, comment and suggestions from the editors and anonymous reviewers, which have substantially improved the paper. In addition, the authors would like to express their special thanks to Ferenc Erdélyi for his useful suggestions and remarks. His unselfish and fruitful cooperation in improving the paper is gratefully acknowledged.

## References

- Albadawi, Z., Bashir, H.A. and Chen, M. (2005) 'A mathematical approach for the formation of manufacturing cells', *Computers & Industrial Engineering*, Vol. 48, No. 1, pp.3–21.
- Alting, L. and Zhang, H. (1989) 'Computer-aided process planning: the state-of-the-art survey', *International Journal of Production Research*, Vol. 27, No. 4, pp.553–585.
- Belohlavek, R., Kulkarni, N. and Vychodil, V. (2009) 'A novel approach to cell formation', in S. Ferr and S. Rudolph (Eds.): *ICFCA 2009, LNAI 5548*, pp.210–223.
- Burbidge, J.I. (1975) *The Introduction to Group Technology*, Halsted Press and John Wiley, New York.
- Busi, M. (2005) *An Integrated Framework for Collaborative Enterprise Performance Management*, PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway.
- Car, Z. and Mikac, T. (2006) 'Evolutionary approach for solving cell-formation problem in cell manufacturing', *Advanced Engineering Informatics*, Vol. 20, No. 3, pp.227–232.
- Chan, H. and Milner, D. (1982) 'Direct clustering algorithm for group formation in cellular manufacturing', *Journal of Manufacturing System*, Vol. 1, No. 1, pp.65–67.
- Chattopadhyay, M., Dan, P.K. and Majumdar, S. (2012) 'Application of visual clustering properties of self organizing map in machine-part cell formation', *Applied Soft Computing*, Vol. 12, No. 2, pp.600–610.
- Folan, P. and Browne, J. (2005) 'A review of performance measurement: towards performance management', *Computers in Industry*, Vol. 56, No. 7, pp.663–680.
- Ganter, B. and Wille, R. (1999) *Formal Concept Analysis: Mathematical Foundations*, Springer Verlag, Berlin-Heidelberg, New York.
- Ganter, B., Stumme, G. and Wille, R. (2005) *Formal Concept Analysis: Foundations and Applications*, Springer Verlag, Berlin-Heidelberg.
- Ganter, B., Körei, A. and Radeleczki, S. (2013) 'Extent partitions and context extensions', *Mathematica Slovaca*, Vol. 63, No. 4, pp.693–706.
- Ham, I., Hitomi, K. and Yoshida, I. (1985) *Group Technology: Applications to Production Management*, Kluwer-Nijhoff Publishing, Boston, Massachusetts.
- Houtzeel, A. (2001) 'Group technology', in Zandin, K.B. (Ed.): *Maynard's Industrial Engineering Handbook*, 5th ed., pp.17.105–17.119, McGraw-Hill, New York.
- Jahanzaib, M., Masood, S. A., Nadeem, S., Akhtar, K. and Shahbaz, M. (2013) 'Application of genetic algorithm (GA) approach in formation of manufacturing cells for group technology', *Life Science Journal*, Vol. 9, No. 4, pp.799–809.
- Kaplan, R.S. and Norton, D.P. (2008) 'Mastering the management system', *Harvard Business Review*, January, Vol. 86, No. 1, pp.63–77.

- Koren, Y. and Shpitalni, M. (2010) 'Design of reconfigurable manufacturing systems' *Journal of Manufacturing Systems*, Vol. 29, No. 4, pp.130–141.
- Kőrei, A. and Radeleczki, S. (2006) 'Box elements in a concept lattice', *Contributions to ICFCA 2006*, Verlag Allgemeine Wissenschaft, pp.41–56.
- Kumar, C. and Chandrasekharan, M.P. (1990) 'Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology', *International Journal of Production Research*, Vol. 28, No. 2, pp.233–243.
- Kusiak, A. (1987) 'The generalized group technology concept', *International Journal of Production Research*, Vol. 25, No. 4, pp.561–569.
- Marri, H.B., Gunasekaran, A. and Grieve, R.J. (1998) 'Computer-aided process planning: a state of the art', *International Journal of Advanced Manufacturing Technology*, Vol. 14, No. 4, pp.261–268.
- Megala, N., Rajendran, C. and Gopalan, R. (2008) 'An ant colony algorithm for cell formation in cellular manufacturing systems', *European Journal of Industrial Engineering*, Vol. 2, No. 3, pp.298–336.
- Mitrofanov, S.P. (1966) *Scientific Principles of Group Technology*, National Lending Library for Science and Technology, Boston.
- Mleczko, J. (2011) 'High-variety products manufacturing based on dynamic classification', *Advances in Manufacturing Science and Technology*, Vol. 35, No. 3, pp.71–86.
- Nanda, J., Simpson, T.W., Kumara, S.R.T. and Shooter, S.B. (2006) 'A methodology for product family ontology development using formal concept analysis and web ontology language', *Journal of Computing and Information Science in Engineering*, Vol. 6, pp.103–113.
- Nanda, J., Thevenot, H.J., Simpson, T.W., Stone, R.B., Bohm, M. and Shooter, S.B. (2007) 'Product family design knowledge representation, aggregation, reuse, and analysis', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 21, No. 2, pp.173–192.
- Niebel, B.W. (1965) 'Mechanized process selection for planning new designs', *ASME 33rd Annual Meeting collected papers*, V6.5 Book 4, Paper No. 737.
- Nouri, H. and Hong, T.S. (2012) 'A bacteria foraging algorithm based cell formation considering operation time', *Journal of Manufacturing Systems*, Vol. 31, No. 3, pp.326–336.
- Powell, D. and Netland, T. (2010) 'Towards an integrated performance measurement system for cellular manufacturing: insights from the case of Volvo Aero Norway', *POMS 21st Annual Conference*, Vancouver, 7–10 May, pp.66–80.
- Scholten, B. (2007) *The Road to Integration. A Guide to Applying the ISA 95 Standard in Manufacturing*, ISA, ISBN 0979 2343 8 7.
- Shang, J.S. and Tadikamalla, P.R. (1998) 'Multicriteria design and control of a cellular manufacturing system through simulation and optimization', *International Journal of Production Research*, Vol. 36, No. 6, pp.1515–1528.
- Srinivasan, G., Narendran, T. and Mahadevan, B. (1990) 'An assignment model for the part families problem in group technology', *International Journal of Production Research*, Vol. 28, No. 1, pp.145–152.
- Studel, H.J. (1984) 'Computer-aided process planning: past, present and future', *International Journal of Production Research*, Vol. 22, No. 2, pp.253–266.
- Tóth, T. and Kőrei, A. (2006) 'New approaches in solving machine-part grouping problems', *Production Systems and Information Engineering* Vol. 3, pp.37–46.
- Veres, L. (2010) *Classification Trees, Rough Sets and Their Applications*, PhD thesis, University of Miskolc, Hungary.

- Xu, X., Wang, L. and Newman, S.T. (2011) 'Computer-aided process planning – a critical review of recent developments and future trends', *International Journal of Computer Integrated Manufacturing*, Vol. 24, No. 1, pp.1–31.
- Yao, Y.Y. and Chen, Y.H. (2006) 'Rough set approximation in formal concept analysis', *Transactions on Rough Sets*, LNCS 4100, pp.285–305.
- Yina, Y. and Yasuda, K. (2005) 'Similarity coefficient methods applied to the cell formation problem: a comparative investigation', *Computers & Industrial Engineering*, Vol. 48, No. 3, pp.471–489.
- Ying, K-C., Lin, Sh-W. and Lu, Ch-Ch. (2011) 'Cell formation using a simulated annealing algorithm with variable neighbourhood', *European Journal of Industrial Engineering*, Vol. 4, No. 1, pp.22–42.