A Novel Approach to Cell Formation*

Radim Belohlavek[1,2], Niranjan Kulkarni[1], and Vilem Vychodil[1,2]

[1] Dept. Systems Science and Industrial Engineering
T. J. Watson School of Engineering and Applied Science
Binghamton University–SUNY, PO Box 6000, Binghamton, NY 13902–6000, USA
rbelohla@binghamton.edu,nkulkar1@binghamton.edu,vychodil@binghamton.edu
[2] Dept. Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic

Abstract. We present an approach to the cell formation problem, known from group technology, which is inspired by formal concept analysis. The cell formation problem consists in allocating parts (objects) to machines (attributes), based on the machine-part matrix. This can be viewed as forming groups consisting of a set of parts and a set of machines. Such groups resemble formal concepts in the input data. Due to the specific nature of the performance assessment in the cell formation problem, good groups can be thought of as rectangles which, unlike those corresponding to formal concepts, contain a few blanks, i.e. which are not full of crosses in terms of formal concept analysis. Moreover, such groups need to be disjoint both in terms of objects and attributes. In this paper, we present an algorithm for the cell formation problem, experimental results, and a comparison to some methods proposed in the literature.

1 Introduction

Group technology (GT) is an approach to manufacturing management which capitalizes on grouping of products with similar manufacturing characteristics. Conceived originally in the 1940s in the Soviet Union, it has since been developed and used in numerous countries [10,15]. There are several benefits from applying GT and they are discussed in e.g. [3,15,18].

One particular application of GT is cellular manufacturing (CM) [2,12,18]. CM involves grouping of machines or processes into manufacturing cells and operation of manufacturing cells. Such grouping is based on parts or part families processed by the machines. This makes CM different from a traditional jobshop environment in which machines are grouped according to their functional similarities [8]. The companies surveyed in [17] reported several benefits from implementing CM, including setup time reduction, material handling cost reduction, equipment cost and labor cost reduction, improvement in quality, improvement in material flow, machine and space utilization, and improvement in employee morale.

* Supported by grant No. 1ET101370417 of GA AV ČR and by institutional support, research plan MSM 6198959214.

S. Ferré and S. Rudolph (Eds.): ICFCA 2009, LNAI 5548, pp. 210–223, 2009.

One of the first problems encountered in implementing CM is the cell formation (CF) problem which consists in grouping of machines into cells, grouping of parts into families, and assignment of the part families to machine cells, so that machine utilization is high and inter-cellular movement is low. Several other constraints need often be considered, such as safety and technological requirements regarding the location of machines, maximum size of cells and maximum number of cells specified by a user, requirements regarding the capacity of machines, or the need for designing flexible cells, but the principal concern is machine utilization and inter-cellular movement [8,9,18].

Several approaches to the CF problem were proposed in the literature. [2,18,12,9,16] provide overviews of these approaches. [13] identifies numerous approaches and classifies them according to their methodology into the following classes:

- descriptive procedures (they include informal methods based on rules of thumb or visual inspection, as well as formal methods based on part coding and classification),
- methods based on cluster analysis (both hierarchical and non-hierarchical clustering algorithms are utilized in these methods),
- methods based on graph partitioning,
- methods based on artificial intelligence techniques (a variety of techniques underlies these approaches, such as rule-based knowledge systems, pattern recognition, and artificial neural networks),
- methods based on mathematical programming (particularly, linear and quadratic programming, and dynamic programming).

The evaluations available in the literature, see e.g. [10,13] suggest that there is no clear winner among the proposed approaches to the CF problem, as the approaches perform differently on different types of datasets.

In this paper, we present a novel approach to the CF problem. The approach is inspired by formal concept analysis (FCA) [5,7]. FCA identifies particular clusters, called formal concepts, in the input data which consists of objects, attributes, and an incidence relation between them. The main idea of our approach consists in linking the conceptual framework of CF to the notions of FCA in a way in which parts correspond to objects, machines correspond to attributes, and the part-machine relationship, indicating which parts need to be processed on which machines, is represented by the incidence relation. Doing so, the formal concepts in the input data obtained using such link can naturally be interpreted as cells whose machines correspond to the attributes of the concept intent and whose parts correspond to the object of the concept extent. In terms of CF, the original methods of FCA can recognize only cells with full machine utilization and allow for overlapping cells. In order to fit the requirements of CF, we thus modify the notions of FCA and develop an algorithm that extracts a set of formal concepts from the part-machine data which can be interpreted as a set of cells provided as a solution to the CF problem. We demonstrate by comparison to other methods described in the literature on several benchmark datasets that our algorithm performs well both in terms of machine utilization and inter-cellular movement.

The paper is organized as follows. In Section 2.1, we define the cell formation problem. Section 2.2 reviews basic notions from formal concept analysis and links them to the conceptual framework of cell formation. Our method for cell formation based on formal concept analysis is presented in Section 3. Section 4 presents an experimental evaluation of the proposed method including a comparison to other methods proposed in the literature. Section 5 contains conclusions and directions for future research.

2 Cell Formation Problem and Formal Concept Analysis

2.1 Cell Formation Problem

Let

$$X = \{M_1, \ldots, M_n\}$$

be a set of machines,

$$Y = \{P_1, \ldots, P_m\}$$

be a set of parts,

$$I \subseteq X \times Y$$

be a machine-part incidence relation with the following interpretation:

$$\langle M, P \rangle \in I \quad \text{iff} \quad \text{part } P \text{ needs to be processed on machine } M.$$

The triplet $\langle X, Y, I \rangle$ can be depicted by a table in which rows correspond to machines, columns correspond to parts, and a table entry is black or white depending on whether $\langle M, P \rangle \in I$ or $\langle M, P \rangle \notin I$. Such table is called a *part-machine matrix* in the cell formation problem.

A *cell* in $\langle X, Y, I \rangle$ is a pair $\langle A, B \rangle$ of a set $A \subseteq X$ of machines and a set $B \subseteq Y$ of parts. The *cell formation problem* (CF problem) can be described as follows.

Definition 1. *A* ***solution*** *to the CF problem is a set*

$$\mathcal{S} = \{\langle A_1, B_1 \rangle, \ldots, \langle A_k, B_k \rangle\} \tag{1}$$

of cells for which

1. $\{A_1, \ldots, A_k\}$ *forms a partition of the set* X *of machines,*
2. $\{B_1, \ldots, B_k\}$ *forms a partition of the set* Y *of parts.*

That is, (1) is a solution if

1. for each $l = 1, \ldots, k$: $A_l \neq \emptyset$ and $B_l \neq \emptyset$,
2. for $i, j = 1, \ldots, k$, $i \neq j$: $A_i \cap A_j = \emptyset$ and $B_i \cap B_j = \emptyset$,
3. $A_1 \cup \cdots \cup A_k = X$ and $B_1 \cup \cdots \cup B_k = Y$.

Given $\langle X, Y, I \rangle$, the following two objectives need to be achieved by any solution $\mathcal{S}$ which is considered to be a "good solution":

1. *high machine utilization* within cells, which means that for each cell $\langle A_l, B_l\rangle \in \mathcal{S}$, the number of machine-part pairs $\langle M_i, P_j\rangle$ in this cell (i.e. pairs $\langle M_i, P_j\rangle \in A_l \times B_l$) for which P_j needs to be processed on M_i (i.e. $\langle M_i, P_j\rangle \in I$) is (relatively) high;
2. *low percentage of exceptional elements*, which means that the number of pairs $\langle M_i, P_j\rangle \in I$ for which M_i belongs to a different cell than P_j (i.e. for each $l = 1, \dots, k$: $\langle M_i, P_j\rangle \notin A_l \times B_l$) is (relatively) low.

Evaluations of goodness of a solution are usually based on some variants of the following functions.

Definition 2. *Given a part-machine matrix represented by* $\langle X, Y, I\rangle$ *and a solution* (1)*, we define*

1. the machine utilization $MU(\mathcal{S})$ *of* $\mathcal{S}$ *by*

$$MU(\mathcal{S}) = \frac{1}{k}\sum_{l=1}^{k}\frac{|(A_l \times B_l) \cap I|}{|A_l| \cdot |B_l|}, \tag{2}$$

2. the percentage of exceptional elements $PE(\mathcal{S})$ *of* $\mathcal{S}$ *by*

$$PE(\mathcal{S}) = \frac{|I - \bigcup_{l=1}^{k} A_l \times B_l|}{m \cdot n}. \tag{3}$$

Given a weight $w \in [0, 1]$*, the* grouping efficiency $GE(\mathcal{S}, w)$ *of* $\mathcal{S}$ *is defined by*

$$GE(\mathcal{S}, w) = w \cdot MU(\mathcal{S}) - (1 - w) \cdot PE(\mathcal{S}). \tag{4}$$

Instead of (3), one sometimes uses

$$PE(\mathcal{S}) = \frac{|I - \bigcup_{l=1}^{k} A_l \times B_l|}{m \cdot n - \sum_{l=1}^{k} |A_l| \cdot |B_l|}. \tag{5}$$

Note that $MU(\mathcal{S})$ is the average machine utilization per cell, given that a machine utilization of a cell is the percentage of entries in a cell which are black. $PE(\mathcal{S})$ given by (3) is the percentage of black entries in the collection of entries which do not belong to any cell.

2.2 Formal Concept Analysis

We refer to [7] and [5] for information on formal concept analysis (FCA). We denote a formal context by $\langle X, Y, I\rangle$, i.e. $I \subseteq X \times Y$ (object-attribute data table, objects $x \in X$, attributes $y \in Y$); the concept-forming operators by $^{\uparrow}$ and $^{\downarrow}$, i.e. for $A \subseteq X$, $A^{\uparrow} = \{y \in Y \mid \text{for each } x \in A : \langle x, y\rangle \in I\}$ and dually for $^{\downarrow}$; a concept lattice of $\langle X, Y, I\rangle$ by $\mathcal{B}(X, Y, I)$, i.e. $\mathcal{B}(X, Y, I) = \{\langle A, B\rangle \in 2^X \times 2^Y \mid A^{\uparrow} = B,\, B^{\downarrow} = A\}$.

3 Proposed Method

This section describes our approach to find a solution of a given cell-formation problem which meets certain quality criteria. The criteria are formulated using suitable measures. The basic idea of our approach can be summarized as follows: (1) Define a function f which measures quality of a cell in a given context; (2) take a formal context $I \subseteq X \times Y$ representing the part-machine relationship; (3) take sets $A \subseteq X$ and $B \subseteq Y$ which maximize f; (4) output $\langle A, B\rangle$ and repeat step (3) until all objects and attributes are covered. Thus, we follow a greedy approach utilizing the measure f.

The greedy approach sketched above may end up in a situation where all machines (parts) are covered by the discovered cells and some of the parts (machines) are not. In such a case, we are not able to form a solution from the discovered cells because one of the conditions 1. and 2. of Definition 1 is not satisfied. We therefore relax the notion of a solution as follows.

Definition 3. *A cell $\langle A, B\rangle$ where exactly one of the sets A and B is empty is called a* ***degenerate cell****. An* ***admissible solution*** *to the CF problem is any set $\mathcal{S} = \{\langle A_1, B_1\rangle, \ldots, \langle A_k, B_k\rangle\}$ of cells such that*
1. *$\mathcal{S}$ contains at most one degenerate cell,*
2. *for $i, j = 1, \ldots, k$, $i \neq j$: $A_i \cap A_j = \emptyset$ and $B_i \cap B_j = \emptyset$, and*
3. *$A_1 \cup \cdots \cup A_k = X$ and $B_1 \cup \cdots \cup B_k = Y$.*

The algorithm can be formalized as follows:

FindCells(I, f)
1 $U \leftarrow X; V \leftarrow Y; \mathcal{C} \leftarrow \emptyset$
2 **while** $U \neq \emptyset$ **and** $V \neq \emptyset$
3 **do** $\langle A, B\rangle \leftarrow$ FindBestCell(I, U, V, f)
4 $\mathcal{C} \leftarrow \mathcal{C} \cup \{\langle A, B\rangle\}; U \leftarrow U - A; V \leftarrow V - B$
5 **if** $U \neq \emptyset$ **or** $V \neq \emptyset$
6 **then**
7 $\mathcal{C} \leftarrow \mathcal{C} \cup \{\langle U, V\rangle\}$;
8 **return** $\mathcal{C}$

The algorithm FindCells(I, f) first initializes sets U and V denoting the remaining objects and attributes which can be used to form cells. FindBestCell(I, U, V, f) at line 3 returns a new cell $\langle A, B\rangle$, i.e. $A \subseteq U$, $B \subseteq V$, which has a high value of f (preferably the highest one) among all possible cells formed from U and V in I. Obviously, FindBestCell(I, U, V, f) can be defined in many ways; some of them will be discussed later. Once a suitable cell $\langle A, B\rangle$ is found by calling FindBestCell(I, U, V, f), objects from A and attributes from B are removed from U and V (see line 4) which ensures that the next cell will not have an overlap with the cells computed in the previous steps. If if-then clause between lines 5–7 adds to $\mathcal{C}$ a degenerate cell $\langle U, V\rangle$ consisting of remanining machines U and parts V provided that $U \neq \emptyset$ or $V \neq \emptyset$. At the end of the computation, $\mathcal{C}$ contains an admissible solution.

Brute-Force Algorithm. We now focus on FindBestCell(I, U, V, f) which is the core of our algorithm. Since FindBestCell is supposed to find a cell which, in the ideal case, maximizes f, the best cell can be obtained by going through all possible subsets of U and V:

FindBestCell$_1(I, U, V, f)$
$s \leftarrow -\infty$
for $C \in 2^U$
  do for $D \in 2^V$
    do if $s < f(I, U, V, C, D)$
      then
        $s \leftarrow f(I, U, V, C, D); \langle A, B\rangle \leftarrow \langle C, D\rangle$
return $\langle A, B\rangle$

Needless to say, such an algorithm has an exponential time complexity. However, it can be applied to some of the small real-world problems presented in the literature.

In what follows we focus on variants of FindBestCell$_1$ which do not go through the space of all possible subsets of U and V but only through a smaller portion which contains promising cells (i.e., cells with high values of f).

Algorithm Using Formal Concepts. The number of cells which are calculated during a single call of FindBestCell$_1(I, U, V, f)$ can be reduced if we use formal concepts as cells. This is based on the idea that any cell $\langle A, B\rangle$ which is a formal concept has a full machine utilization and it is a maximal cell containing $\langle A, B\rangle$ with this property. This is due to the fact that formal concepts in I correspond to maximal rectangles in I containing black entries only. The corresponding modification of FindBestCell$_1$ can be formalized as follows:

FindBestCell$_2(I, U, V, f)$
1 $s \leftarrow -\infty$
2 **for** $\langle C, D\rangle \in \mathcal{B}(X, Y, I)$
3   **do** $C \leftarrow C \cap U; D \leftarrow D \cap V$
4   **if** $s < f(I, U, V, C, D)$
5   **then**
6         $s \leftarrow f(I, U, V, C, D); \langle A, B\rangle \leftarrow \langle C, D\rangle$
7 **return** $\langle A, B\rangle$

Note that after obtaining a formal concept $\langle C, D\rangle \in \mathcal{B}(X, Y, I)$ at line 2, the sets C and D are restricted to the objects and attributes from the sets of remaining objects U and attributes V only. (See line 3.) As we demonstrate in the next section, in several cases this method can deliver results which are nearly as good as the results obtained by the brute-force algorithm. In addition to that, if the best possible solution to the cell-formation problem exists, solution $\mathcal{S}$ for which $MU(\mathcal{S}) = 1$ and $PE(\mathcal{S}) = 0$, it will be always found:

Theorem 1. *If the cell-formation problem for $I \subseteq X \times Y$ has a solution such that $MU(\mathcal{S}) = 1$ and $PE(\mathcal{S}) = 0$, this solution can be found by* FindCells *combined with* FindBestCell$_2$.

Proof. Let f be a function such that $f(I, U, V, C, D) = 1$ if (i) $(C \times D) \cap I = C \times D$ (i.e., if $\langle C, D\rangle$ is a rectangle full of black entries) and (ii) $((U - C) \times D) \cap I = \emptyset$ and

$(C\times(V-D))\cap I=\emptyset$ (i.e., both $(U-C)\times D$ and $C\times(V-D)$ are empty rectangles), and $f(I,U,V,C,D)<1$ otherwise. (Such assumptions can be considered natural requirements for f.) It is easily seen that if $MU(\mathcal{S})=1$ and $PE(\mathcal{S})=0$, then for each cell $\langle C,D\rangle$ in the solution of the cell-formation problem, we have $f(I,X,Y,C,D)=1$ and $\langle C,D\rangle$ needs to be a maximal rectangle. Therefore, the first cell generated by the algorithm is one the cells contained in the solution. The rest follows directly by induction. □

Algorithm Using Dense Rectangles. If the best possible solution does not exist, i.e., if there is no admissible solution for a cell-formation problem such that $MU(\mathcal{S})=1$ and $PE(\mathcal{S})=0$, the method based on formal concepts may not yield optimal results. Intuitively, there may be interesting cells which have almost full machine utilization (i.e., $MU(\mathcal{S})$ is close to 1) and low percentage of exceptional elements. Such cell may be more useful than a cell with full machine utilization but a higher percentage of exceptional elements.

Therefore, we modify the "cells as concepts" approach to include cells formed of rectangles almost full of 1's. We find such rectangles by finding formal concepts first and then adding promising objects and attributes as long as the quality measure of the particular rectangle increases. This leads to a modified version of the algorithm described in the previous paragraph:

```
FindBestCell3(I, U, V, f)
s ← −∞
for ⟨E, F⟩ ∈ B(X, Y, I)
   do E ← E ∩ U; F ← F ∩ V
      repeat
          C ← E; D ← F; r ← f(I, U, V, C, D)
          select M ∈ U − C that maximizes f(I, U, V, C ∪ {M}, D)
          select P ∈ V − D that maximizes f(I, U, V, C, D ∪ {P})
          if f(I, U, V, C ∪ {M}, D) ≤ f(I, U, V, C, D ∪ {P})
             then q ← f(I, U, V, C, D ∪ {P}); F ← D ∪ {P};
             else  q ← f(I, U, V, C ∪ {M}, D); E ← C ∪ {M};
        until q < r
      if s < f(I, U, V, C, D)
        then
              s ← f(I, U, V, C, D); ⟨A, B⟩ ← ⟨C, D⟩
return ⟨A, B⟩
```

Compared to the previous algorithm, a new repeat∼until loop is added. The loop searches for remaining objects and attributes which can be added to the current rectangle (originally, a formal concept) and which increase (or do not decrease) the quality measure.

Quality Measures. The quality of solution found by FindCells combined with any of the variants of FindBestCell depends on our choice of the quality measure f. For given I, U, V, and $A\subseteq U$ and $B\subseteq V$, $f(I,U,V,A,B)$ is the measure of goodness of $\langle A,B\rangle$ in I. We consider quality measures which assign higher values to better cells. Although the notion of a "better cell" is subjective, we can agree that in certain situations there are "best cells" with full machine utilization (in the cell) and no exceptional elements (in the cell).

All the quality measures proposed below use percentages of machine utilization and exceptional elements in a single cell to compute the resulting value of f. We introduce functions $g(I, U, V, A, B) \in [0, 1]$ and $h(I, U, V, A, B) \in [0, 1]$ as follows:

$$g(I, U, V, A, B) = \frac{|(A \times B) \cap I|}{|A| \cdot |B|}, \tag{6}$$

$$h(I, U, V, A, B) = 1 - \frac{|((A \times (V - B)) \cup ((U - A) \times B)) \cap I|}{|A| \cdot |V - B| + |U - A| \cdot |B|}. \tag{7}$$

Clearly, $g(I, U, V, A, B)$ is the machine utilization of $\langle A, B\rangle$, i.e. the fraction of black entries in the rectangle $\langle A, B\rangle$. Analogously, $h(I, U, V, A, B)$ is the fraction of non-exceptional elements, i.e. the fraction of white entries in the rectangles $\langle A, V - B\rangle$ and $\langle U - A, B\rangle$. Obviously, if $\langle A, B\rangle$ is a cell with full machine utilization and no exceptional elements then $g(I, U, V, A, B) = h(I, U, V, A, B) = 1$.

Remark 1. In the rest of this section, we fix I, U, V and simplify the notation: For brevity, we write just $f(A, B)$, $g(A, B)$, $h(A, B), \ldots$ instead of $f(I, U, V, A, B)$, $g(I, U, V, A, B)$, $h(I, U, V, A, B), \ldots$

We can introduce two families of quality measures which are based on weighted arithmetic and geometric averages of $g(A, B)$ and $h(A, B)$. Namely, we define f_1^w and f_2^w as follows:

$$f_1^w(A, B) = w \cdot g(A, B) + (1 - w) \cdot h(A, B), \tag{8}$$

$$f_2^w(A, B) = g(A, B)^w \cdot h(A, B)^{\frac{1}{w}}. \tag{9}$$

These measures will be used in the next section.

4 Experimental Evaluation

In this section we present examples of solutions to sample cell-formation problems which we identified in the literature and provide a comparison with other approaches. First, let us note that qualifying a solution as "good" among admissible solutions is highly subjective. Usually, an expert judgment or additional knowledge is needed to select the best solution among several ones. Second, despite the computational complexity of the method we propose, the results are obtained with acceptable response times because the data sets that appear in the cell-formation problem domain are usually small (with $|Y|$ around 30 or less). In this section we focus on the effect of selecting various quality measures and variants of the FindBestCell algorithm.

Results Obtained by Variants of FindBestCell *Algorithm.* As discussed in the previous section, the algorithm can deliver the best solution possible if it exists. If not, the algorithm varies based on the choice of f and FindBestCell$_i$. If one

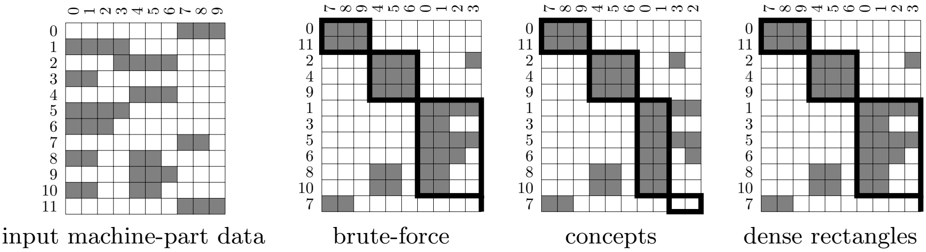

Fig. 1. Various quality measures used to solve the same cell-formation problem

uses FINDBESTCELL$_2$ using formal concepts as cells, we usually get an acceptable solution which however might not be natural for a user. For instance, in Fig. 1 we have the input data matrix (left), a brute-force admissible solution (second from left), and a solution found by taking "formal concepts as cells" (third from left) with f_2^1 as the quality measure. Note that the cells in Fig. 1 are depicted by thick rectangles drawn in data tables with permuted rows and columns. For example, the first cell in the second diagram consists of machines 0 and 11 and parts 7, 8, and 9.

For an industrial engineer, the second solution may seem not natural because it has more exceptional elements and it contains an "empty cell" with no machine utilization. Interestingly, if we use FINDBESTCELL$_3$ which uses "dense rectangles", we obtain an admissible solution which is the same is in case of the brute-force algorithms (first from right). We have tested the algorithms on various artificial as well as real-world machine-part datasets and we have observed that the admissible solution using "dense rectangles" produces the same or almost as good results as the brute-force algorithm with considerable smaller demands.

Choice of Quality Measures. The choice of a quality measure, i.e. function f, seems to be crucial for finding a satisfactory solution. There seems to be no single measure which works well for all datasets because the problem of finding a "good solution" is subjective. In general, good choices seem to be $f_1^{0.5}$ (equal emphasis is put on utilization and exceptions) and f_2^w with lower values of w (tends to suppress exceptions). Sample results corresponding to various measures are depicted in Fig. 2.

In Fig. 2, we have used the "dense rectangles" to form the cells. Notice that f_1^0 is trivial because all emphasis is put on no exceptional elements with no emphasis on the utilization, i.e. a trivial solution is to have one cell covering the entire part-machine dataset. In a similar sense, f_1^1 produces degenerate solution as well because the high demand of utilization leaves large amount of exceptional elements and 4 machines are not contained in any cell. In case of this dataset, $f_2^{0.01}$ seems to produce the best solution.

Illustrative Examples. We now show examples of (admissible) solutions to the cell-formation problem with datasets which we identified in the literature [1,6,14]. The solutions have been found using "dense rectangles" as cells and f_2^w taken

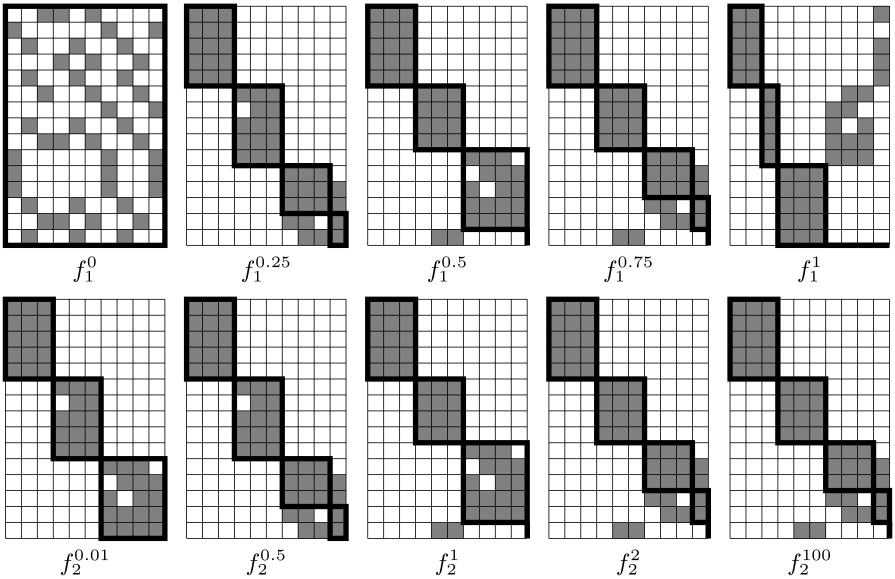

Fig. 2. Various quality measures used to solve the same cell-formation problem

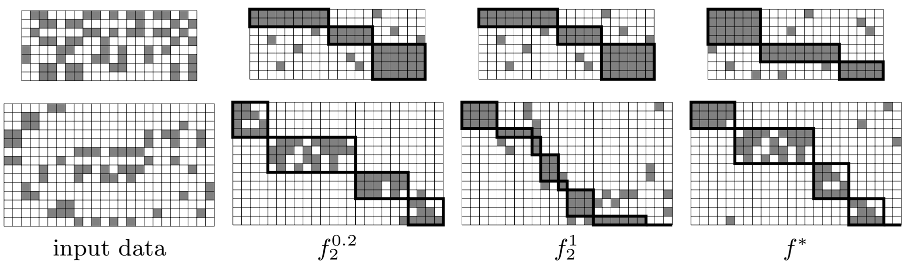

Fig. 3. Various quality measures used to solve the same cell-formation problem

as the quality measure. In addition to that, we have used a quality measure which takes into account the size of cells. The motivation is the following: an engineer often follows not only the utilization but also the numbers of cells. As an extreme example, it does not make much sense to have as much cells as machines in the system, making each machine a separate cell. Therefore, we introduce the following measure:

$$f^*(A,B) = g(A,B) \cdot h(A,B)^2 \cdot \ln\Big(1 + \sqrt{|A| \cdot |B|}\Big), \tag{10}$$

which is similar to f_2^w except for it multiplies the result by the size of a possible cell, putting more emphasis on large cells. In order to avoid another extreme (having all machines in one or just a few cells), we adjusted the quality measure by the logarithm of the size of the edge of a possible cell. Fig. 3 contains results for two datasets using various quality measures.

Table 1. Comparing results from [14] with the proposed method for the 8×20 matrix from Fig. 3 (top)

Method	Cells	Cell 1 Utilization	Cell 2 Utilization	Cell 3 Utilization	Exceptional Elements	Voids
[14]	3	1	1	1	9	0
$f_2^{0.2}$	3	1	1	1	9	0
f_2^1	3	1	1	1	9	0
f^*	3	1	1	1	9	0

Table 2. Comparing results from [1] with the proposed method for 12×10 matrix from Fig. 1

Method	Cells	Cell 1 Utilization	Cell 2 Utilization	Cell 3 Utilization	Cell 4 Utilization	Exceptional Elements	Voids
[1]	3	0.8125	0.8667	0.8889	-	5	6
Brute Force	3	1	1	0.7083	-	7	7
Formal Concepts	4	1	1	1	0	12	0
Dense Rectangles	3	1	1	0.7083	-	7	7

In the first example, the measures produce practically the same solution which can be seen as satisfactory solutions. In the second case, $f_2^{0.2}$ and f^* produce similar solutions which have lower machine utilization than the solution obtained using f_2^1 (which has full machine utilization). On the contrary, f_2^1 has a large number of small cells and larger amount of exceptional elements. Which of the solutions is actually the best one depends on particular application and preferences of users. This example demonstrates that by tuning parameters of quality functions, one can influence the solutions based on user-specified requests (i.e., larger utilization, smaller number of cells, etc.).

Comparison With Other Approaches. The aim of this section is to compare the quality results obtained by other authors to the results obtained by the method proposed in this paper. For this purpose, we use some of the datasets which we identified in the literature. We present the comparisons by means of uniform tables which contain the characteristics of solutions available in the literature.

The data sets used for comparisons are benchmark data problems and are obtained from [1,6,14]. We use the comparison criteria for which the results are available in those papers. The papers use different approaches to solve the cell formation problem. [1] proposes a two-phase approach. The first phase makes use of principal component analysis to identify machine cells; the second phase uses an integer programming model to assign parts to these identified machine cells. [6] uses a particular iterative clustering algorithm to find cells. [14] utilizes a similarity matrix assessing similarity between machines and uses this matrix in an assignment procedure which solves a particular maximization problem to form cells.

Table 3. Comparing results from [14] with the proposed method for 14 × 24 matrix from Fig. 2 (bottom)

Method	Cells	Cell 1 Utilization	Cell 2 Utilization	Cell 3 Utilization	Cell 4 Utilization	Cell 5 Utilization	Cell 6 Utilization	Cell 7 Utilization	Exceptional Elements	Voids
[14]	4	1	0.5	0.625	0.639	-	-	-	4	29
$f_2^{0.2}$	4	0.6875	0.6	0.8333	0.6667	-	-	-	2	28
f_2^1	7	1	1	1	1	1	1	1	20	0
f^*	4	0.9333	0.6389	0.6875	0.6667	-	-	-	4	23

Table 4. Comparing results from [6] with the proposed method for 15 × 10 matrix from Fig. 2

Method	Cells	Cell 1 Utilization	Cell 2 Utilization	Cell 3 Utilization	Cell 4 Utilization	Exceptional Elements	Voids
[6]	3	0.8125	1	0.9333	-	0	4
$f_2^{0.01}$	3	1	0.9333	0.85	-	0	4
f_2^1	4	1	0.9333	1	1	6	1
f_2^1	3	1	1	0.85	-	2	3

Table 1, Table 2, Table 3, and Table 4 contain the comparisons. For every table, we provide a reference to the paper from which we got the dataset and the characteristics of the solutions obtained by the authors in the respective paper. For every solution listed, we provide the number of cells in the solution, machine utilization for every cell, the number of exceptional elements (black entries in the data matrix which are not covered by any cell), and the number of voids (white entries in the cells).

5 Conclusions

We presented a new method for the cell-formation problem known from the group technology. The method is inspired by formal concept analysis. We provided results of experiments and a basic comparison with other methods presented in the literature. One advantage of our method is that it is transparent in that it does not use any preprocessing method (such as the principal component analysis) which some of the methods in the literature use. Another advantage is the fact that our method is parameterizable and yields solutions based on user's preference regarding the importance of machine utilization and exceptional elements, as well as the overall number of cells.

Future research will include more comprehensive comparison with existing approaches; exploring the possibility to add interactivity to the method via a visual inspection of the concept lattice associated to the input data (the user might give some initial information to the algorithm based on such inspection); and extending our method to be able to take into account in a natural way the user's expert knowledge and preferences. In addition, we plan to explore other approaches to dense rectangles which appeared in the literature (suggested by an anonymous reviewer).

Acknowledgment

We thank the anonymous reviewers for their helpful comments.

References

1. Albadawi, Z., Bashir, H.A., Chen, M.: A mathematical approach for the formation of manufacturing cells. Computers & Industrial Engineering 48, 3–21 (2005)
2. Askin, R.G., Selim, H.M., Vakharia, A.J.: A methodology for designing flexible cellular manufacturing systems. IIE Transactions 29(7), 599–610 (1997)
3. Belarmino, A.-D., Lozano, S., Racero, J., Guerrero, F.: Machine cell formation in generalized group technology. Computers & Industrial Engineering 41(2), 227–240 (2001)
4. Belohlavek, R., Vychodil, V.: Dense rectangles in object-attribute data. In: Proc. IEEE GrC 2006, IEEE International Conference on Granular Computing, Atlanta, GA, May 10–12, 2006, pp. 586–591 (2006) IEEE Catalog Number 06EX1286, ISBN 1-4244-0133-X
5. Carpineto, C., Romano, G.: Concept Data Analysis. Theory and Applications. Wiley, Chichester (2004)
6. Chan, H., Milner, D.: Direct clustering algorithm for group formation in cellular manufacturing. J. Manufacturing Systems 1(1), 65–67 (1982)
7. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer, Heidelberg (1999)
8. Heragu, S.S.: Group technology and cellular manufacturing. IEEE Transactions on Systems, Man, and Cybernetics 24(2), 203–215 (1994)
9. Khan, M.S.I., Bhaba, S.R.: A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems. International Journal of Production Research 38(3), 699–720 (2000)
10. Miltenburg, J., Zhang, W.: A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology. Journal of Operations Management 10(1), 44–72 (1991)
11. Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine 2(6), 559–572 (1901)
12. Pillai, M.V., Subbarao, K.: A robust cellular manufacturing system design for dynamic part population using a genetic algorithm. International Journal of Production Research 46(18), 5191–5210 (2008)
13. Selim, H., Askin, R.G., Vakharia, A.: Cell Formation in Group Technology: Review, Evaluation, and Direction for Future Research. Computers & Industrial Engineering 34(1), 3–20 (1998)

14. Srinivasan, G., Narendran, T., Mahadevan, B.: An assignment model for the part families problem in group technology. Int. J. Production Research 28(1), 145–152 (1990)
15. Suresh, N.C., Meredith, J.R.: Achieving factory automation through group technology principles. Journal of Operations Management 5(2), 151–167 (1985)
16. Vakharia, A.J.: Method of cell formation in group technology: a framework for evaluation. Journal of Operations Management 6(3-4), 257–271 (1986)
17. Wemmerlöv, U., Hyer, N.L.: Cellular manufacturing in the U.S. industry: A survey of users. Int. Journal of Production Research 27(9), 1511–1530 (1989)
18. Yasuda, K., Yin, Y.: A dissimilarity measure for solving the cell formation problem in cellular manufacturing. Computers & Industrial Engineering 39, 1–17 (2001)