Computers and Operations Research 91 (2018) 112–120

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An efficient exact model for the cell formation problem with a variable number of production cells

Ilya Bychkov*, Mikhail Batsyn

Laboratory of Algorithms and Technologies for Network Analysis, National Research University Higher School of Economics, 136 Rodionova, Nizhniy Novgorod, 603093, Russian Federation

ARTICLE INFO

Article history:
Received 7 April 2016
Revised 27 August 2017
Accepted 13 November 2017
Available online 16 November 2017

Keywords:
Cell formation problem
Cellular manufacturing
Fractional objective
Two-index model
Grouping efficacy

ABSTRACT

The Cell Formation Problem has been studied as an optimization problem in manufacturing for more than 90 years. It consists of grouping machines and parts into manufacturing cells in order to maximize loading of cells and minimize movement of parts from one cell to another. Many heuristic algorithms have been proposed which are doing well even for large-sized instances. However, only a few authors have aimed to develop exact methods and most of these methods have some major restrictions such as a fixed number of production cells for example. In this paper we suggest a new mixed-integer linear programming model for solving the cell formation problem with a variable number of manufacturing cells. The popular grouping efficacy measure is used as an objective function. To deal with its fractional nature we apply the Dinkelbach approach. Our computational experiments are performed on two testsets: the first consists of 35 well-known instances from the literature and the second contains 32 instances less popular. We solve these instances using CPLEX software. Optimal solutions have been found for 63 of the 67 considered problem instances and several new solutions unknown before have been obtained. The computational times are greatly decreased comparing to the state-of-art approaches.

1. Introduction

The Cell Formation Problem as a part of Group Technology (GT) was introduced by Burbidge (1961) and Mitrofanov (1966). In the most general formulation it is designed to reduce production costs by grouping machines and parts into manufacturing cells (production shops). The goal of such kind of grouping is to set up manufacturing process in a way that maximizes loading of machines within the cells and minimizes movement of parts from one cell to another. In classical formulation the problem is defined by a binary matrix A with m rows representing machines and p columns representing parts. In this machine-part matrix $a_{ij} = 1$ if part j is processed on machine i. The objective is to form production cells, which consist of machines and parts together, optimizing some production metrics such as machine loading and intercell movement.

As an example of input data we will consider the instance of Waghodekar and Sahu (1984) shown in Table 1. This instance consists of 5 machines and 7 parts. The ones in a machine-part matrix are called *operations*. In Table 2 a solution with 2 manufacturing cells is presented. The first manufacturing cell contains machines m_1, m_4 with parts p_1, p_7 and the second manufacturing cell contains machines m_2, m_3, m_5 with parts p_2, p_3, p_4, p_5, p_6. Some parts have to be moved from one cell to another for processing (e.g. part p_6 needs to be processed on machine m_1, so it should be transported from its cell 2 to cell 1). The operations lying outside cells are called *exceptional elements* or *exceptions*. There can be also non-operation elements inside cells ($a_{ij} = 0$). These elements reduce machine load and are called *voids*. So the goal is to minimize the number of exceptions and the number of voids at the same time.

1.1. Related work

Many different approaches are proposed for solving the cell formation problem. The majority of them provide heuristic solutions and only a few exact methods have been suggested.

Krushinsky and Goldengorin (1984) provided two MINpCUT exact models based on the well-known k-cut graph partition problem. The objective function considered in this research is minimization of the exceptional elements number for a fixed number of cells. Unfortunately this objective function does not address the load inside cells.

Elbenani and Ferland (2012) presented a mixed-integer linear programming model which maximizes the most popular objective for the cell formation problem - the grouping efficacy, introduced

[*] Corresponding author.
E-mail addresses: ibychkov@hse.ru (I. Bychkov), mbatsyn@hse.ru (M. Batsyn).

https://doi.org/10.1016/j.cor.2017.11.009
0305-0548/© 2017 Elsevier Ltd. All rights reserved.

Table 1
Machine-part 5 × 7 matrix from Waghodekar and Sahu (1984).

	p_1	p_2	p_3	p_4	p_5	p_6	p_7
m_1	1	0	0	0	1	1	1
m_2	0	1	1	1	1	0	0
m_3	0	0	1	1	1	1	0
m_4	1	1	1	1	0	0	0
m_5	0	1	0	1	1	1	0

Table 2
Solution with 2 production cells.

	p_7	p_1	p_6	p_5	p_4	p_3	p_2
m_1	1	1	1	1	0	0	0
m_4	0	1	0	0	1	1	1
m_2	0	0	0	1	1	1	1
m_3	0	0	1	1	1	1	0
m_5	0	0	1	1	1	0	1

by Kumar and Chandrasekharan (1990). These authors suggested to apply Dinkelbach algorithm since the grouping efficacy is a fractional objective function. Their model allows solving the cell formation problem only if the number of production cells is predefined. Thus the suggested approach cannot guarantee global optimality of the obtained solutions with respect to a variable number of production cells. In many cases the computational times for this model are quite long or memory limitations are exceeded and the optimal solutions cannot be found.

Brusco (2015) introduced two approaches for solving the cell formation problem with the grouping efficacy objective. The first is a mixed-integer linear programming model which is based on a general two-mode clustering formulation with some simplifying assumptions (e.g. the numbers of clusters by rows and columns are equal). This model looks interesting, but requires too much time to be solved for many even medium-sized instances. The second approach is a branch-and-bound algorithm combined with a relocation heuristic to obtain an initial solution. The branch and bound approach is able to solve about two times more problem instances and the computational times are greatly improved as well. Generally it runs fine on well-structured (with grouping efficacy value $>$ 0.65–0.7) medium-sized problems. Two major assumptions are made for both of these approaches: singletons are permitted (manufacturing cells containing only one machine or one part) that is quite a common practice; residual cells are permitted (cells containing only machines without parts, or only parts without machines). Also the number of production cells is predefined for the both approaches, but for some test instances several values for the number of cells are considered in computational experiments.

Another model is provided in our earlier paper (Bychkov et al., 2014). There we present a mixed-integer linear programming formulation for the cell formation problem with a variable number of production cells. It deals well with small-sized instances, but nevertheless the number of variables and constraints is huge - $O(m^2p)$. This does not allow obtaining solutions even for some moderate-sized test instances and in some cases this model runs too slowly.

Some authors used biclustering approaches to solve the cell formation problem. Boutsinas (2013) applied simultaneous clustering for both dimensions (machines and parts) and minimized the number of voids plus the number of exceptional elements. Pinheiro et al. (2016) reduced the cell formation problem to another biclustering problem - bicluster graph editing problem and suggested an exact method and a linear programming model which provides good computational results for the grouping efficacy objective.

Table 3
Testset A - instances.

ID	Source	m	p
A1	King and Nakornchai (1982) - Fig. 1a	5	7
A2	Waghodekar and Sahu (1984) - Problem 2	5	7
A3	Seifoddini (1989b)	5	18
A4	Kusiak and Cho (1992)	6	8
A5	Kusiak and Chow (1987)	7	11
A6	Boctor (1991) - Example 1	7	11
A7	Seifoddini and Wolfe (1986)	8	12
A8	Chandrasekaran and Rajagopalan (1986a)	8	20
A9	Chandrasekaran and Rajagopalan (1986b)	8	20
A10	Mosier and Taube (1985a)	10	10
A11	Chan and Milner (1982)	15	10
A12	Askin and Subramanian (1987)	14	24
A13	Stanfel (1985)	14	24
A14	McCormick et al. (1972)	16	24
A15	Srinivasan et al. (1990)	16	30
A16	King (1980)	16	43
A17	Carrie (1973)	18	24
A18	Mosier and Taube (1985b)	20	20
A19	Kumar et al. (1986)	23	20
A20	Carrie (1973)	20	35
A21	Boe and Cheng (1991)	20	35
A22	Chandrasekharan and Rajagopalan (1989) - Dataset 1	24	40
A23	Chandrasekharan and Rajagopalan (1989) - Dataset 2	24	40
A24	Chandrasekharan and Rajagopalan (1989) - Dataset 3	24	40
A25	Chandrasekharan and Rajagopalan (1989) - Dataset 5	24	40
A26	Chandrasekharan and Rajagopalan (1989) - Dataset 6	24	40
A27	Chandrasekharan and Rajagopalan (1989) - Dataset 7	24	40
A28	McCormick et al. (1972)	27	27
A29	Carrie (1973)	28	46
A30	Kumar and Vannelli (1987)	30	41
A31	Stanfel (1985) - Fig. 5	30	50
A32	Stanfel (1985) - Fig. 6	30	50
A33	King and Nakornchai (1982)	30	90
A34	McCormick et al. (1972)	37	53
A35	Chandrasekharan and Rajagopalan (1987)	40	100

1.2. Contributions of this research

In this paper we develop a fast compact model providing optimal solutions for the cell formation problem with a variable number of manufacturing cells and the grouping efficacy objective. Unlike the majority of linear programming models our model does not contain a direct assignment of machines or parts to cells. We use machine-machine and part-machine assignments instead of the widely used machine-part-cell assignment. This leads to a compact and elegant formulation considering only constraints which ensure a block-diagonal structure of solutions. It allows us to drastically reduce the number of variables and constraints in our programming model and obtain globally optimal solutions even for some large-sized problem instances.

Computational experiments show that our model outperforms all known exact methods. We have solved 63 of 67 problem instances to the global optimum with respect to a variable number of production cells. We have also found several new solutions unknown before.

We would like to highlight that many researchers in the field use the 35 GT instances dataset provided by Gonçalves and Resende (2004). These instances are taken from different cell formation research papers (references to the original sources are shown in Table 3). Some problem instances in this 35 GT dataset have errors and differ from the ones presented in the original papers. Many researchers including Elbenani and Ferland (2012) and Pinheiro et al. (2016) have performed their computational experiments using these data from Gonçalves and Resende (2004). We have reviewed all the original sources, comparing and forming the corrected version of this popular dataset. We have also collected many other problem instances less popular and formed a new dataset. All data can be downloaded from website opt-hub.com or researchgate.net (full urls can be found in references).

The paper is organized as follows. In Section 2 we provide the formulation of the cell formation problem. Then in Section 3 we present our new mixed-integer linear programming model. Sections 4 contains the information about datasets we used for our experiments and the computational results and comparisons to other exact approaches are given in Section 5. The conclusion is provided in Section 6.

2. Problem formulation

Cellular manufacturing systems apply are aimed to process similar parts within the same production cell, balance machines workload and minimize parts movement from one cell to another during the production process. The most popular objective for the cell formation problem is the grouping efficacy introduced by Kumar and Chandrasekharan (1990):

$$\tau = \frac{n_1^{in}}{n_1 + n_0^{in}}$$

where

n_1 – the total number of operations (ones) in the machine-part matrix,

n_1^{in} – the number of operations performed inside cells,

n_0^{in} – the number of voids (zeros inside cells).

In comparison to the other objectives the grouping efficacy function addresses the best block-diagonal structure of the cell formation problem solutions (Sarker, 2001).

In the literature several constraints related to the minimal size of a cell could be found. The following are the three most popular considerations:

- Allowing residual cells (cells containing only machines or parts).
- Allowing singletons (cells with one machine and several parts or vice versa) and prohibiting residual cells.
- Allowing only cells with at least 2 machines and 2 parts.

The most popular option is allowing singletons and prohibiting residual cells. In this section for the classical formulation we assume that singletons can appear in solutions and residual cells are prohibited. In our new model and in computational experiments we consider the first two options.

A straightforward integer fractional programming (IFP) model for the cell formation problem with the grouping efficacy objective function allowing singletons and prohibiting residual cells is given below. We use the following notation: m is the number of machines, p is the number of parts, a_{ij} equals to 1 if machine i processes part j and c is the maximal possible number of production cells. Since each production cell has to contain at least one machine and at least one part we set $c = \min(m, p)$.

(IFP model):

Decision variables:

$$x_{ik} = \begin{cases} 1, & \text{if machine } i \text{ belongs to cell } k, \\ 0, & \text{otherwise} \end{cases}$$

$$y_{jk} = \begin{cases} 1, & \text{if part } j \text{ belongs to cell } k, \\ 0, & \text{otherwise} \end{cases}$$

$$max \quad \frac{\sum_{i=1}^{m}\sum_{j=1}^{p}\sum_{k=1}^{c} a_{ij}x_{ik}y_{jk}}{\sum_{i=1}^{m}\sum_{j=1}^{p} a_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{p}\sum_{k=1}^{c}(1-a_{ij})x_{ik}y_{jk}} \qquad (1)$$

Subject to:

$$\sum_{k=1}^{c} x_{ik} = 1 \quad i = 1, \ldots, m \qquad (2)$$

$$\sum_{k=1}^{c} y_{jk} = 1 \quad j = 1, \ldots, p \qquad (3)$$

$$\sum_{i=1}^{m} x_{ik} \le m \cdot \sum_{j=1}^{p} y_{jk} \quad k = 1, \ldots, c \qquad (4)$$

$$\sum_{j=1}^{p} y_{jk} \le p \cdot \sum_{i=1}^{m} x_{ik} \quad k = 1, \ldots, c \qquad (5)$$

Objective function (1) is the grouping efficacy measure where the numerator is the number of ones inside cells (n_1^{in}) and two sums in the denominator are the total number of ones (n_1) and the number of zeros inside cells (n_0^{in}) respectively. Constraints (2) and (3) require that each machine and each part is assigned to exactly one production cell. The following two inequalities (4) and (5) prohibit residual cells (without machines or parts). The left part of (4) is the total number of machines assigned to the particular cell (this sum is not greater than m) and the right part is the total number of parts assigned to that cell (multiplied by m). It means that if we have at least one machine assigned to some cell there should be at least one part assigned to this cell. This model allows us to have any number of cells in the optimal solution. For example if optimal solution has only two cells then variables x_{ik} and y_{jk} will be zero for all k except only two values of k.

3. MILP model

3.1. Objective linearization

In our paper Bychkov et al. (2014) we have proposed a mixed-integer linear programming model for the cell formation problem which is very similar to the one described in the previous section. One of the most important points there was linearization of the grouping efficacy objective. Our previous idea was to linearize the grouping efficacy objective function by fixing the value of denominator $n_1 + n_0^{in}$ and considering a range of all possible numbers of voids n_0^{in}. The lower bound for n_0^{in} equals to 0 because generally there can be a solution without any voids. The upper bound is computed using the following proposition.

Proposition 1 (Bychkov et al., 2014). *The number of voids in the optimal solution satisfies the following inequality:*

$$n_0^{in} \le \left\lfloor \frac{1-\tau}{\tau} n_1 \right\rfloor$$

where τ is the grouping efficacy value of any feasible solution.

So if we know a feasible solution we can limit the range of possible values for the number of voids. Unfortunately, the performance of this approach strongly depends on the feasible solution we use for obtaining our bounds. This way solving problem instances where grouping efficacy value is low takes too much computational resources (since the number of subtasks is too large) and sometimes we are unable to solve even medium-sized cell formation instances.

In this paper together with using our new mixed-integer linear model we use another way of linearization – Dinkelbach (1967) algorithm. The parametric approach introduced by Dinkelbach is one of the most general and popular strategies for fractional programming. It reduces the solution of a fractional programming problem to the solution of a sequence of simpler problems. If we consider a fractional programming model with the following objective function:

$$Q(x) = \frac{P(x)}{D(x)}, \qquad (6)$$

then Dinkelbach procedure is the following:

- **Step 1** Take some feasible solution x^0, compute $\lambda_1 = \frac{P(x^0)}{D(x^0)}$ and let $k = 1$
- **Step 2** Solve the original problem with objective function $Q(x)$ replaced with $F(\lambda_k) = P(x) - \lambda_k D(x) \rightarrow$ max and let x^k be the optimal solution
- **Step 3** If $F(\lambda_k)$ is equal to 0 (or less than some predefined tolerance) then stop the procedure and return x^k as the optimal solution.
  Else $k = k+1$, $\lambda_k = \frac{P(x^k)}{D(x^k)}$ and goto step 2.

Elbenani and Ferland (2012) have also used Dinkelbach approach for linearization of grouping efficacy measure. Although their computational times are quite high and the results are given only for the particular fixed number of production cells.

3.2. Suggested two-index model

Due to a large number of variables and constraints in three-index model (Bychkov et al., 2014) CPLEX spends too much computational resources solving even small-sized instances (we have solved only 14 of 35 problem instances). Here we introduce a two-index mixed-integer linear programming model for the cell formation problem. The key idea of this model is removing machine-part-cell relation as it has been done in many works before. Instead of mapping elements to cells we use a simple fact that machines within the same production cell have the same set of parts assigned to that cell. The two-index model combines well with the Dinkelbach algorithm and shows impressing performance even on large-sized problem instances.

Two-index model:

$$x_{ik} = \begin{cases} 1, & \text{if machines } i \text{ and } k \text{ are in the same cell,} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if machine } i \text{ and part } j \text{ are in the same cell,} \\ 0, & \text{otherwise} \end{cases}$$

$$max \quad \sum_{i=1}^{m}\sum_{j=1}^{p} a_{ij}y_{ij} - \lambda \cdot \left(\sum_{i=1}^{m}\sum_{j=1}^{p}(1-a_{ij})y_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{p} a_{ij} \right) \tag{7}$$

Subject to:

$$2x_{ik} - y_{ij} - y_{kj} \geq -1 \quad i,k = 1,\dots,m \quad j = 1,\dots,p \tag{8}$$

$$y_{ij} - y_{kj} - x_{ik} \geq -1 \quad i,k = 1,\dots,m \quad j = 1,\dots,p \tag{9}$$

$$y_{kj} - y_{ij} - x_{ij} \geq -1 \quad i,k = 1,\dots,m \quad j = 1,\dots,p \tag{10}$$

$$\sum_{j=1}^{p} y_{ij} \geq 1 \quad i = 1,\dots,m \tag{11}$$

$$\sum_{i=1}^{m} y_{ij} \geq 1 \quad j = 1,\dots,p \tag{12}$$

Technically matrix $[x_{ik}]$ here can be replaced by the one with part-part relations, however the number of machines in problem instances is usually lower than the number of parts (for large-sized instances the difference is significant). As a result we have m^2 variables from matrix $[x_{ik}]$ and mp variables from matrix $[y_{ij}]$.

Objective function (7) is the grouping efficacy measure linearized using Dinkelbach method. Constraints (8)–(10) set relations between machines and parts to ensure the solution can be transformed into the block-diagonal form (which means its feasibility).

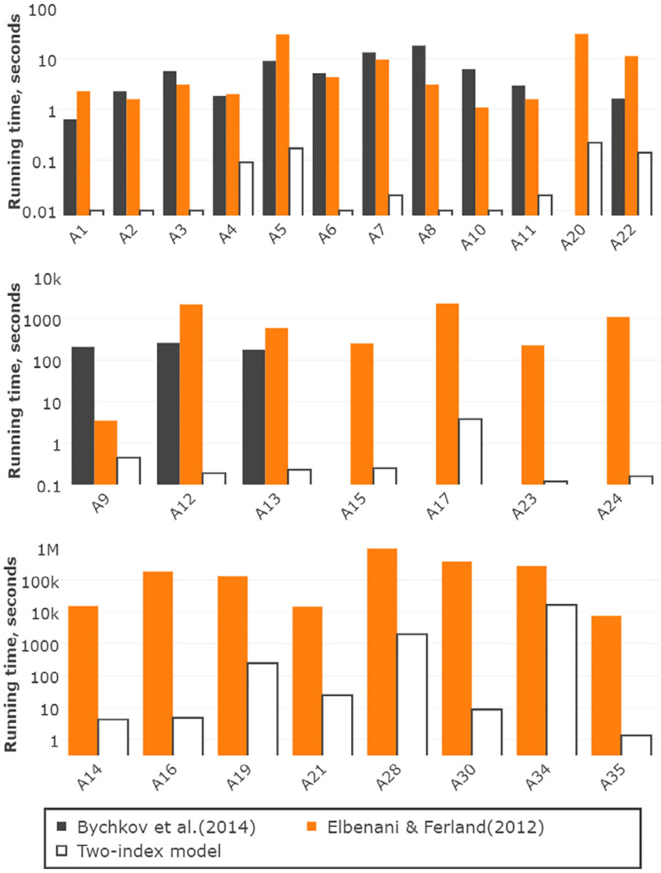

Fig. 1. Testset A - No residual cells. Running times comparison.

The last two inequalities (11) and (12) are optional and prohibit residual cells.

We start with setting λ equal to the best known efficacy value for the considered cell formation problem instance. Then we sequentially solve several two-index problems according to the Dinkelbach algorithm and update λ value with the solutions found until our objective function is above 0.

4. Test instances

For our computational experiments we have used two datasets, *Testset A* and *Testset B*.

Testset A - Classic. The first dataset is a classical 35 GT problem set taken from Gonçalves and Resende (2004). It contains 35 test instances with sizes from 5 × 7 up to 40 × 100 (machines × parts notation). This dataset is very popular among cell formation researchers and there are lots of computational results obtained by different methods (heuristics and metaheuristics generally). As we highlighted before some problem instances in this dataset have inconsistencies with the original papers they are published in. We have compared these instances to the original ones and corrected the dataset.

Testset B - Extra. Another dataset named *Testset B* consists of other instances taken from different papers. We have looked through many papers on the cell formation problem and formed this new set. There are 32 test instances with sizes from 6 × 6 to 50 × 150. A couple of instances from this set have been adopted to the classical formulation of the cell formation problem.

Since the number of machines determines the size of our model (the number of decision variables and constraints) we consider 3 classes of problem instances.

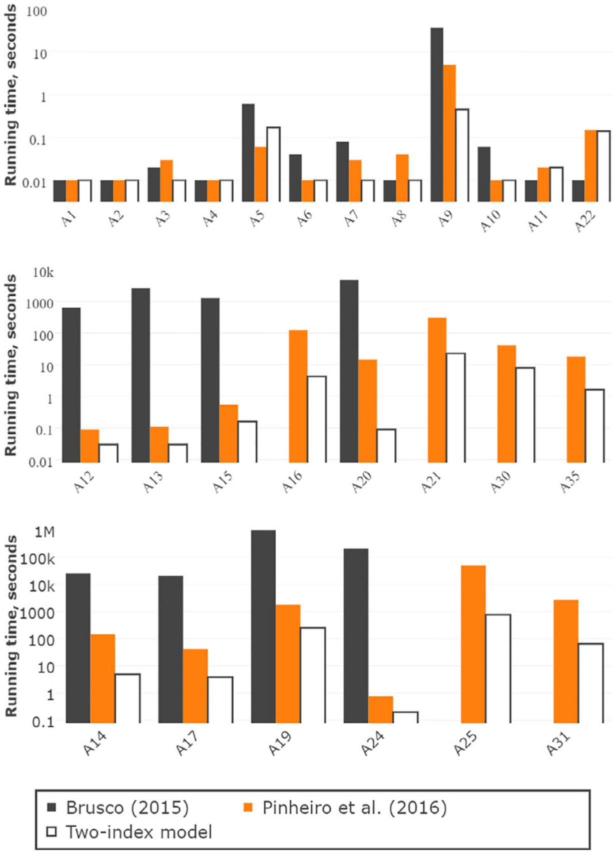

Fig. 2. Testset A - Allowed residual cells. Running times comparison.

Table 4
Testsets instances size.

	Small	Medium	Large
Testset A	9	8	18
Testset B	11	13	8

- Small (less than 10 machines).
- Medium (from 10 to 20 machines).
- Large (20 machines or greater).

For our data we can conclude that Testset A has 2 times more large instances, but less medium and small instances (see Table 4).

5. Computational results

For our computational experiments we consider two most popular versions of cell size constraints:

1. Residual cells are prohibited, singletons are allowed (each cell has at least 1 machine and 1 part).
2. Residual cells are allowed (cells with only machines or only parts can appear in the final solution).

Several state-of-art exact approaches have been chosen for comparisons. As a platform for our computations we have used a system with Intel Xeon processor running at 3.4 GHz with 16GB RAM and CPLEX 12.4.0 installed. Due to high-quality initial solutions the Dinkelbach algorithm makes only one or, in rare cases, two iterations.

Table 5
Testset B - instances.

ID	Source	m	p
B1	Adil et al. (1996)	6	6
B2	Pa Rkin and Li (1997)	6	7
B3	Brown and Sumichrast (2001)	6	11
B4	Chan and Milner (1982)	7	5
B5	Kusiak and Chow (1987)	7	8
B6	Zolfaghari and Liang (2002)	7	8
B7	Won and Kim (1997)	7	10
B8	Sarker and Khan (2001)	8	8
B9	Nair (1999)	8	10
B10	Islam and Sarker (2000)	8	10
B11	Kumar et al. (1986)	9	15
B12	Ham et al. (1985)	10	8
B13	Viswanathan (1996)	10	12
B14	Shargal et al. (1995)	10	38
B15	Won and Kim (1997)	11	10
B16	Seifoddini (1988)	11	22
B17	Moon and Chi (1992)	12	19
B18	Li (2003)	14	14
B19	Chan and Milner (1982) - Fig. 3a	15	10
B20	Yang and Yang (2008) - Fig. 6b	15	15
B21	Yang and Yang (2008) - Fig. 6c	15	15
B22	Yang and Yang (2008) - Fig. 6d	15	15
B23	Harhalakis et al. (1994)	17	20
B24	Seifoddini and Djassemi (1991)	18	24
B25	Sandbothe (1998)	20	10
B26	Nagi et al. (1990)	20	51
B27	Won and Kim (1997)	26	28
B28	Yang and Yang (2008) - Fig. 7	28	35
B29	Seifoddini and Djassemi (1996)	35	15
B30	Seifoddini and Djassemi (1996)	41	50
B31	Yang and Yang (2008) - Fig. 12	46	105
B32	Zolfaghari and Liang (1997)	50	150

5.1. *Testset A*

5.1.1. *Experiments*

The instances from Table 3 have been studied widely in the literature. We report results separately for the formulation where minimal cell size is 1 × 1 (Table 7 and Fig. 1) and the formulation with residual cells allowed (Table 8 and Fig. 2). In the first case we have selected two approaches for the results comparison:

1. The MILP model by Elbenani and Ferland (2012).
2. The MILP model by Bychkov et al. (2014).

Elbenani and Ferland (2012) considered a simplified formulation of the cell formation problem solving every problem instance only for one fixed number of production cells. These authors have performed computational experiments on an AMD processor 2.2 GHz with 4GB RAM. For Testset A we use the best efficacy results from the literature as initial values for λ parameter.

In case of unrestricted cell sizes (residual cells are allowed) we have compared our results to the following approaches:

1. The branch-and-bound algorithm by Brusco (2015).
2. The ILP model by Pinheiro et al. (2016).
3. The iterative exact method by Pinheiro et al. (2016).

Brusco (2015) considers several values for the number of cells for some problem instances, so in this case we compare our computational time with these times summed up for every test instance. As hardware platforms Brusco (2015) reports 3.4 GHz Intel Core i7-2600 with 8GB RAM and Pinheiro et al. (2016) the same 3.4 GHz Intel Core i7-2600 with 32 GB RAM.

Since Elbenani and Ferland (2012) and Brusco (2015) do not consider all possible numbers of production cells we show the number of cells in brackets for these approaches.

Table 6
Computational experiments on the data provided by Gonçalves and Resende (2004).

#	Time, s		Efficacy	
	Pinheiro et al. (2016)	Two-index	Pinheiro et al. (2016)	Two-index
A1	0.01	0.01	0.7500	0.7500
A7	0.03	0.01	0.6944	0.6944
A14	144.91	4.99	0.5333	0.5333
A15	0.54	0.17	0.6992	0.6992
A17	42.32	3.51	0.5773	0.5773
A20	14.55	0.11	0.7938	0.7938
A21	305.48	15.08	0.5879	0.5879
A25	48743.90	678.53	0.5329	0.5329
A30	41.53	8.58	0.6304	0.6304

Table 7
Testset A - computational results (residual cells are prohibited, singletons are allowed).

#	Time, s			Efficacy		
	Elbenani and Ferland (2012)	Bychkov et al. (2014)		Elbenani and Ferland (2012)	Bychkov et al. (2014)	Two-index model
A1	2.3	0.63	0.00	0.8235(2)	0.8235	0.8235
A2	1.6	2.29	0.00	0.6957(2)	0.6957	0.6957
A3	3.1	5.69	0.00	0.7959(2)	0.7959	0.7959
A4	2.0	1.86	0.09	0.7692(2)	0.7692	0.7692
A5	30.6	9.14	0.17	0.6087(5)	0.6087	0.6087
A6	4.3	5.15	0.01	0.7083(4)	0.7083	0.7083
A7	9.6	13.37	0.02	0.6944(4)	0.6944	0.6944
A8	3.1	18.33	0.01	0.8525(3)	0.8525	0.8525
A9	3.5	208.36	0.45	0.5872(2)	0.5872	0.5872
A10	1.1	6.25	0.00	0.7500(5)	0.7500	0.7500
A11	1.6	2.93	0.02	0.9200(3)	0.9200	0.9200
A12	2188.7	259.19	0.19	0.7206(7)	0.7206	0.7206
A13	593.2	179.21	0.23	0.7183(7)	0.7183	0.7183
A14	15130.5	*	4.24	0.5326(8)	*	**0.5326**
A15	252.5	*	0.25	0.6953(6)^{E}	*	**0.6899**
A16	183232.5	*	4.80	0.5753(8)	*	**0.5753**
A17	2345.6	*	3.82	0.5773(9)	*	**0.5773**
A18	*	*	32243.10	*	*	**0.4345**
A19	131357.5	*	245.59	0.5081(7)	*	**0.5081**
A20	31.1	*	0.22	0.7791(5)	*	**0.7791**
A21	14583.6	*	24.34	0.5798(5)	*	**0.5798**
A22	11.3	1.64	0.14	1.0000(7)	1.0000	1.0000
A23	230.7	*	0.12	0.8511(7)	*	**0.8511**
A24	1101.1	*	0.16	0.7351(7)	*	**0.7351**
A25	*	*	1026.96	*	*	**0.5329**
A26	*	*	178182.24	*	*	**0.4895**
A27	*	*	*	*	*	*
A28	958714.1	*	1964.00	0.5482(5)	*	**0.5482**
A29	*	*	*	*	*	*
A30	378300.0	*	8.72	0.6331(14)	*	**0.6331**
A31	*	*	136.00	0.6012(13)^{E}	*	**0.5977**
A32	*	*	*	*	*	*
A33	*	*	*	*	*	0.4800
A34	268007.6	*	16323.71	0.6064(3)	*	**0.6064**
A35	7365.3	*	1.34	0.8403(10)	*	**0.8403**

5.1.2. Results

The results for Testset A are summarized in Tables 7 and 8. For each algorithm we report the grouping efficacy value and the running time in seconds. Since our testset is larger than the one used by Brusco (2015) the missing results are marked as "–". For some problems exact solutions have not been obtained because CPLEX runs too long or takes too much memory. These instances are marked as "*".

Table 7 shows the results for the case where we prohibit cells without machines or parts. Our previous model from Bychkov et al. (2014) also considers a variable number of production cells, but due to its complexity and not very effective linearization technique it is able to solve only 14 test problems of 35. The model suggested by Elbenani and Ferland (2012) solved 27 problem instances but only for the one fixed number of production cells for each problem instance. Our new model provides global optimal solutions (with respect to any possible number of cells) for 31 of 35 problem instances. For problem instance A33 we have found a new solution with grouping efficacy 0.48 unknown before.

For 17 instances: A14–A21, A23–A26, A28, A30, A31, A34 and A35 we are the first to prove the global optimality of the best known solutions with respect to a variable number of production cells.

Running times bar chars for Table 7 are presented in Fig. 1. Here we have used logarithmic scale with base 10 for Y axis (running time). Our new model shows really good performance, it works from 7 to 43,383 times faster than the model from Elbenani and Ferland (2012) and from 11 to 1833 times faster than the model from Bychkov et al. (2014). We must underline that although we use a better hardware platform than Elbenani and Ferland (2012), our problem formulation is more complicated than a formulation with a fixed number of cells.

Table 8
Testset A - computational results (residual cells are allowed).

#	Time, s				Efficacy		
	Brusco (2015)	Pinheiro et al. (2016)	Pinheiro et al. (2016)	Two-index	Brusco (2015)	Pinheiro et al. (2016)	Two-index
A1	0.01	0.16	0.01	0.01	0.8235(2,3,4)	0.7500^E	0.8235
A2	0.01	0.07	0.01	0.01	0.6957(2,3,4)	0.6956	0.6957
A3	0.02	0.09	0.03	0.01	0.8085(2,3,4)	0.8085	0.8085
A4	0.01	0.02	0.01	0.01	0.7916(2,3,4)	0.7917	0.7917
A5	0.6	0.29	0.06	0.17	0.6087(2,3,4,5,6)	0.6087	0.6087
A6	0.04	0.14	0.01	0.01	0.7083(2,3,4,5)	0.7083	0.7083
A7	0.08	0.18	0.03	0.01	0.6944(2,3,4,5)	0.6944^E	0.6944
A8	0.01	2.06	0.04	0.01	0.8525(2,3,4)	0.8525	0.8525
A9	35.86	81.46	4.94	0.45	0.5872(2,3,4)	0.5872	0.5872
A10	0.06	0.03	0.01	0.01	0.7500(2,3,4,5,6)	0.7500	0.7500
A11	0.01	0.01	0.02	0.02	0.9200(2,3,4)	0.9200	0.9200
A12	633.91	0.49	0.09	0.03	0.7424(6,7,8)	0.7424	0.7424
A13	2631.76	0.49	0.11	0.03	0.7285(6,7,8)	0.7286	0.7286
A14	24716.34	600.98	144.91	4.88	0.5385(8)	0.5333^E	0.5385
A15	1279.93	7.24	0.54	0.16	0.6992(5,6,7)	0.6992^E	0.6992
A16	–	1156.23	125.62	4.24	–	0.5804	0.5804
A17	20840.55	87.13	42.32	3.84	0.5773(9)	0.5773^E	0.5773
A18	–	*	*	52810.10	–	*	**0.4397**
A19	1375608.66	23928.70	1771.99	249.52	0.5081(7)	0.5081	0.5081
A20	4830.00	1.78	14.55	0.09	0.7888(5,6,7)	0.7938^E	0.7888
A21	–	2145.24	305.48	22.60	–	0.5879^E	0.5860
A22	0.01	0.02	0.15	0.14	1.0000(7)	1.0000	1.0000
A23	42.29	10.08	0.44	0.14	0.8511(7)	0.8511	0.8511
A24	208158.02	17.46	0.78	0.20	0.7351(7)	0.7351	0.7351
A25	–	371233.00	48743.90	759.70	–	0.5329^E	0.5329
A26	–	*	*	134418.65	–	*	**0.4895**
A27	–	*	*	*	–	*	*
A28	–	*	*	46361.97	–	*	**0.5482**
A29	–	*	*	*	–	*	*
A30	–	183.71	41.53	8.00	–	0.6304^E	0.6331
A31	–	13807.50	2622.06	64.82	–	0.5977	0.5977
A32	–	*	*	234055.90	–	*	**0.5084**
A33	–	*	*	*	–	*	0.4829
A34	–	*	*	14212.57	–	*	**0.6131**
A35	–	325.53	18.22	1.61	–	0.8403	0.8403

The results for the formulation with no constraints on cell sizes are summarized in Table 8. The model suggested by Pinheiro et al. (2016) solved 27 problem instances to the global optimum. Our approach has obtained exact solutions for 32 of 35 test instances. In addition for problem instances A18, A33 and A34 we have found new solutions unknown before.

Running times bar charts for Table 8 are presented in Fig. 2. Here we have chosen the ILP model from Pinheiro et al. (2016) for comparison since it has a better performance than the exact iterative method of the same authors. In Fig. 2 we have also used logarithmic scale with base 10 for the first and second plots (instances with running times less than 60 s and less than 5000 s). For the last plot (instances with running times less than 1,500,000 s) we have used logarithmic scale with base 100. We can see that the two-index model runs up to 1 million times faster than the branch-and-bound algorithm by Brusco (2015) and up to 161 times faster than the ILP model by Pinheiro et al. (2016).

5.1.3. Inconsistencies

The classical dataset of 35 GT problems from Gonçalves and Resende (2004) have been used for many years by the cell formation researchers for computational experiments and results comparison. Unfortunately, the dataset contains several inconsistencies with the original sources: papers from King and Nakornchai (1982) to Chandrasekharan and Rajagopalan (1987) (see Table 3). Many researchers have used corrupted instances and sometimes add some new inconsistencies. Therefore obtaining results for these problems and comparing it to results of other approaches becomes a really difficult task. One of the goals of this paper is to provide correct data for the cell formation researchers. In this paper we mark usage of inconsistent data with superscript ^{E}.

We have not been able to obtain results reported in Elbenani and Ferland (2012) for problem instances A15 and A31 using both possible data sets - dataset from Gonçalves and Resende (2004) and our corrected version. Probably some other data have been used.

Several instances provided by Gonçalves and Resende (2004), which are different from its original sources (papers from King and Nakornchai, 1982 to Chandrasekharan and Rajagopalan, 1987, see Table 3), have been also used by Pinheiro et al. (2016). These instances are A1, A7, A14, A15, A17, A20, A21, A25 and A30. For a fair comparison we have also run our model using the same input data (see Table 6). Our experiments have confirmed all the results obtained by Pinheiro et al. (2016). Also we can conclude that the running times of our model have not changed much on these input data.

5.2. Testet B results

Since the test instances from Table 5 are less popular in research papers our goal is just to obtain optimal solutions for this set. We have used our multi-start local search heuristic (Bychkov et al., 2013) to get good solutions which are then passed as initial values for λ parameter (we pick the best solution found by the heuristic within 30 s).

The results for Testset B are shown in Table 9. Here we have found optimal solutions for 31 of 32 test problems. Another result is an excellent performance of our multi-start local search heuristic algorithm: only one of 32 instances solved by the heuristic differs

Table 9
Testset B - computational results.

#	Time		Efficacy		
	Two-index (no residual cells)	Two-index (allow residual cells)	Heuristic bound	Two-index (no residual cells)	Two-index (allow residual cells)
B1	0.01	0.01	0.8095	0.8095	0.8095
B2	0.01	0.01	0.7222	0.7222	0.7222
B3	0.25	0.03	0.6071	0.6071	0.6071
B4	0.01	0.01	0.8889	0.8889	0.8889
B5	0.01	0.01	0.7500	0.7500	0.7500
B6	0.01	0.01	0.7391	0.7391	0.7391
B7	0.01	0.01	0.8148	0.8148	0.8148
B8	0.01	0.01	0.7222	0.7222	0.7222
B9	0.01	0.01	0.7576	0.7576	0.7576
B10	0.01	0.01	0.9000	0.9000	0.9000
B11	0.01	0.02	0.7273	0.7273	0.7297
B12	0.01	0.01	0.8276	0.8276	0.8276
B13	0.36	0.80	0.5962	0.5962	0.6042
B14	0.25	0.30	0.6405	0.6405	0.6405
B15	0.01	0.01	0.8333	0.8333	0.8333
B16	0.16	0.06	0.7391	0.7391	0.7444
B17	0.98	0.26	0.6552	0.6552	0.6842
B18	1.82	1.65	0.6027	0.6129	0.6129
B19	0.03	0.06	0.8000	0.8000	0.8113
B20	0.05	0.03	0.8710	0.8710	0.8710
B21	0.03	0.04	0.8333	0.8333	0.8333
B22	0.05	0.01	0.7258	0.7258	0.7258
B23	0.05	0.06	0.8111	0.8111	0.8111
B24	4.79	7.80	0.5673	0.5673	0.5728
B25	0.20	0.10	0.7600	0.7600	0.8000
B26	13.81	25.75	0.6068	0.6068	0.6078
B27	0.25	0.28	0.7248	0.7248	0.7248
B28	0.83	1.04	0.6729	0.6729	0.6729
B29	33.82	51.76	0.5730	0.5730	0.5745
B30	4.76	8.67	0.7308	0.7308	0.7325
B31	19.69	17.50	0.6799	0.6799	0.6799
B32	*	*	0.6193	*	*

from the exact solution (instance B18). Due to the high computational complexity we are unable to solve the largest problem in the set – problem B32 (50 × 150).

6. Conclusion

The cell formation problem is a well known combinatorial optimization problem with a high computational complexity. A very few authors have suggested exact approaches for the most popular problem formulation with the grouping efficacy objective function. The majority of these works assume that the number of production cells is predefined. In this paper we suggest a new compact and effective integer linear programming model for the cell formation problem with a variable number of production cells. The model is based on the machine-machine and part-machine relations instead of the widely used machine-part-cell relation. It allows us to drastically reduce the number of variables and constraints in the resulting integer linear program. Computational experiments show that our new model outperforms the state-of-art exact methods. We have solved 63 of 67 problem instances to the global optimum with respect to a variable number of production cells. We have also found several new solutions unknown before. Unfortunately many problem instances from the cell formation datasets have inconsistencies with the original papers. This makes it really difficult to perform computational experiments and compare results to other approaches in the field. We have extracted and checked over 67 problem instances. All these data are available for downloading from website opt-hub.com or researchgate.net and we hope it will help the researchers in this area. The suggested model can be also used for solving biclustering problems and this is one of the directions of our future work.

Acknowledgments

This work was conducted at National Research University Higher School of Economics, Laboratory of Algorithms and Technologies for Network Analysis and supported by RSF grant 14-41-00039.

References

Adil, G.K., Rajamani, D., Strong, D., 1996. Cell formation considering alternate routeings. Int. J. Prod. Res. 34 (5), 1361–1380.
Askin, R.G., Subramanian, S., 1987. A cost-based heuristic for group technology configuration. Int. J. Prod. Res. 25 (1), 101–113.
Boctor, F., 1991. A jinear formulation of the machine-part cell formation problem. Int. J. Prod. Res. 29 (2), 343–356.
Boe, W.J., Cheng, C.H., 1991. A close neighbour algorithm for designing cellular manufacturing systems. Int. J. Prod. Res. 29 (10), 2097–2116.
Boutsinas, B., 2013. Machine-part cell formation using biclustering. Eur. J. Oper. Res. 230, 563–572.
Brown, E.C., Sumichrast, R.T., 2001. CF-GGA: a grouping genetic algorithm for the cell formation problem. Int. J. Prod. Res. 39 (16), 3651–3669.
Brusco, M.J., 2015. An exact algorithm for maximizing grouping efficacy in part-machine clustering. IIE Trans. 47 (6), 653–671.
Burbidge, J.L., 1961. The new approach to production. Prod. Eng. December, 3-19.
Bychkov, I., Batsyn, M., Sukhov, P., Pardalos, P.M., 2013. Heuristic algorithm for the cell formation problem. In: Models, Algorithms, and Technologies for Network Analysis. In: Springer Proceedings in Mathematics & Statistics, 59, pp. 43–69.
Bychkov, I.S., Batsyn, M.V., Pardalos, P.M., 2014. Exact model for the cell formation problem. Optim. Lett. 8 (8), 2203–2210.
Carrie, A.S., 1973. Numerical taxonomy applied to group technology and plant layout. Int. J. Prod. Res. 11, 399–416.
Chan, H.M., Milner, D.A., 1982. Direct clustering algorithm for group formation in cellular manufacture. J. Manuf. Syst. 1 (1), 64–76.
Chandrasekaran, M.P., Rajagopalan, R., 1986a. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. Int. J. Prod. Res. 24, 451–463.
Chandrasekaran, M.P., Rajagopalan, R., 1986b. MODROC: an extension of rank order clustering of group technology. Int. J. Prod. Res. 24 (5), 1221–1233.
Chandrasekharan, M.P., Rajagopalan, R., 1987. ZODIAC - an algorithm for concurrent formation of part families and machine cells. Int. J. Prod. Res. 25 (6), 835–850.

Chandrasekharan, M.P., Rajagopalan, R., 1989. Groupability: an analysis of the properties of binary data matrices for group technology. Int. J. Prod. Res. 27 (6), 1035–1052.

Dinkelbach, W., 1967. On nonlinear fractional programming. Manage. Sci. 13 (7), 492–498.

Elbenani, B., Ferland, J.A., 2012. Cell Formation Problem Solved Exactly with the Dinkelbach Algorithm. CIRRELT-2012-07. Montreal. Quebec., pp. 1–14.

Gonçalves, J.F., Resende, M.G.C., 2004. An evolutionary algorithm for manufacturing cell formation. Comput. Ind. Eng. 47 (2–3), 247–273.

Ham, I., Hitomi, K., Yoshida, T., 1985. Layout planning for group technology. In: Group Technology. In: International Series in Management Science/Operations Research, pp. 153–169.

Harhalakis, G., Ioannou, G., Minis, I., Nagi, R., 1994. Manufacturing cell formation under random product demand. Int. J. Prod. Res. 32 (1), 47–64.

Islam, K.M.S., Sarker, B.R., 2000. A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems. Int. J. Prod. Res. 38 (3), 699–720.

King, J.R., 1980. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. Int. J. Prod. Res. 18 (2), 213–232.

King, J.R., Nakornchai, V., 1982. Machine-component group formation in group technology: review and extension. Int. J. Prod. Res. 20 (2), 117–133.

Krushinsky, D., Goldengorin, B., 1984. An exact model for cell formation in group technology. Comput. Manag. Sci. 9 (3), 323–338.

Kumar, K.R., Chandrasekharan, M.P., 1990. Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. Int. J. Prod. Res. 28 (2), 233–243.

Kumar, K.R., Kusiak, A., Vannelli, A., 1986. Grouping of parts and components in flexible manufacturing systems. Eur. J. Oper. Res. 24, 387–397.

Kumar, K.R., Vannelli, A., 1987. Strategic subcontracting for efficient disaggregated manufacturing. Int. J. Prod. Res. 25 (12), 1715–1728.

Kusiak, A., Cho, M., 1992. Similarity coefficient algorithm for solving the group technology problem. Int. J. Prod. Res. 30, 2633–2646.

Kusiak, A., Chow, W., 1987. Efficient solving of the group technology problem. J. Manuf. Syst. 6 (2), 117–124.

Li, M.L., 2003. The algorithm for integrating all incidence matrices in multi-dimensional group technology. Int. J. Prod. Econ. 86, 121–131.

McCormick, W.T., Schweitzer, P.J., White, T.W., 1972. Problem decomposition and data reorganization by a clustering technique. Oper. Res. 20 (5), 993–1009.

Mitrofanov, S.P., 1966. The Scientific Principles of Group Technology. National Lending Library Translation, Boston Spa, Yorkshire. Translation of Mitrofanov (1959).

Moon, Y.B., Chi, S.C., 1992. Generalized part family formation using neural network techniques. J. Manuf. Syst. 11 (3), 149–159.

Mosier, C.T., Taube, L., 1985a. The facets of group technology and their impact on implementation. OMEGA 13 (6), 381–391.

Mosier, C.T., Taube, L., 1985b. Weighted similarity measure heuristics for the group technology machine clustering problem. OMEGA 13 (6), 577–583.

Nair, G.J., 1999. Accord: a Bicriterion algorithm for cell formation using ordinal and ratio-level data. Int. J. Prod. Res. 37 (3), 539–556.

Pa Rkin, R.E., Li, M.-L., 1997. The multi-dimensional aspects of a group technology algorithm. Int. J. Prod. Res. 35 (8), 2345–2358.

Pinheiro, R.G.S., Martins, I.C., Protti, F., Ochi, L.S., Simonetti, L.G., Subramanian, A., 2016. On solving manufacturing cell formation via bicluster editing. Eur. J. Oper. Res. 254 (3), 769–779.

Nagi, R., Harhalakis, G., Proth, J.-M., 1990. Multiple routings and capacity considerations in group technology applications. Int. J. Prod. Res. 28 (12), 1243–1257.

Sandbothe, R.A., 1998. Two observations on the grouping efficacy measure for goodness of block diagonal forms. Int. J. Prod. Res. 36 (11), 3217–3222.

Sarker, B.R., 2001. Measures of grouping efficiency in cellular manufacturing systems. Eur. J. Oper. Res. 130, 588–611.

Sarker, B.R., Khan, M., 2001. A comparison of existing grouping efficiency measures and a new weighted grouping efficiency measure. IIE Trans. 33, 11–27.

Seifoddini, H., 1988. Comparison between single linkage and average linkage clustering techniques in forming machine cells. Comput. Indus. Eng. 15, 210–216.

Seifoddini, H., 1989b. A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. Int. J. Prod. Res. 27 (7), 1161–1165.

Seifoddini, H., Djassemi, M., 1991. The production data-based similarity coefficient versus Jaccard's similarity coefficient. Comput. Indus. Eng. 21, 263–266.

Seifoddini, H., Djassemi, M., 1996. The threshold value of a quality index for formation of cellular manufacturing systems. Int. J. Prod. Res. 34 (12), 3401–3416.

Seifoddini, H., Wolfe, P.M., 1986. Application of the similarity coefficient method in group technology. IIE Trans. 18 (3), 271–277.

Shargal, M., Shekhar, S., Irani, S.A., 1995. Evaluation of search algorithms and clustering efficiency measures for machine-part matrix clustering. IIE Trans. 27 (1), 43–59.

Srinivasan, G., Narendran, T.T., Mahadevan, B., 1990. An assignment model for the part-families problem in group technology. Int. J. Prod. Res. 28 (1), 145–152.

Stanfel, L., 1985. Machine clustering for economic production. Eng. Costs Prod. Econ. 9, 73–81.

Viswanathan, S., 1996. A new approach for solving the p-median problem in group technology. Int. J. Prod. Res. 34 (10), 2691–2700.

Waghodekar, P.H., Sahu, S., 1984. Machine-component cell formation in group technology MACE. Int. J. Prod. Res. 22, 937–948.

Won, Y., Kim, S., 1997. Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings. Comput. Ind. Eng. 32 (1), 207–220.

Yang, M.S., Yang, J.H., 2008. Machine-part cell formation in group technology using a modified ART1 method. Eur. J. Oper. Res. 188, 140–152.

Zolfaghari, S., Liang, M., 1997. An objective-guided ortho-synapse hopfield network approach to machine grouping problems. Int. J. Prod. Res. 35 (10), 2773–2792.

Zolfaghari, S., Liang, M., 2002. Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine–grouping problems. Int. J. Prod. Res. 40 (9), 2141–2158.