

# Flexible PMP Approach for Large-Size Cell Formation

**Boris Goldengorin**

LATNA–Laboratory of Algorithms and Technologies for Networks Analysis and Department of Higher Mathematics,  
The National Research University Higher School of Economics, Moscow 101000, Russia; and  
Department of Mathematics and Informatics, Khmel'nitsky University of Economics and Business,  
Khmel'nitsky 29000, Ukraine, [bgoldengorin@hse.ru](mailto:bgoldengorin@hse.ru)

**Dmitry Krushinsky, Janes Slomp**

Department of Operations, University of Groningen, 9700 AV Groningen, The Netherlands  
{[d.krushinsky@gmail.com](mailto:d.krushinsky@gmail.com), [j.slomp@rug.nl](mailto:j.slomp@rug.nl)}

Lately, the problem of cell formation (CF) has gained a lot of attention in the industrial engineering literature. Since it was formulated (more than 50 years ago), the problem has incorporated additional industrial factors and constraints while its solution methods have been constantly improving in terms of the solution quality and CPU times. However, despite all the efforts made, the available solution methods (including those for a popular model based on the  $p$ -median problem, PMP) are prone to two major types of errors. The first error (the modeling one) occurs when the intended objective function of the CF (as a rule, verbally formulated) is substituted by the objective function of the PMP. The second error (the algorithmic one) occurs as a direct result of applying a heuristic for solving the PMP. In this paper we show that for instances that make sense in practice, the modeling error induced by the PMP is negligible. We exclude the algorithmic error completely by solving the adjusted pseudo-Boolean formulation of the PMP exactly, which takes less than one second on a general-purpose PC and software. Our experimental study shows that the PMP-based model produces high-quality cells and in most cases outperforms several contemporary approaches.

*Subject classifications:* cell formation;  $p$ -median problem; pseudo-Boolean polynomial; group technology.

*Area of review:* Optimization.

*History:* Received February 2011; revisions received February 2012, June 2012; accepted July 2012.

## 1. Introduction

Cell formation (CF), a popular concept in industrial engineering, suggests grouping machines into manufacturing cells and parts into product families so that each family is processed mainly within one cell. The problem of optimal (usually with respect to the volume of intercell movement) cell formation has been studied by many researchers. An overview can be found in Selim et al. (1998), Yin and Yasuda (2006), and recently in Bhatnagar and Saddikuti (2010). The  $p$ -median problem (PMP) was frequently used in the past as a modeling tool for cell formation problems (Kusiak and Chow 1987, Wang and Roze 1997, Deutsch et al. 1998, Won and Lee 2004, Ashayeri et al. 2005, Won and Currie 2006). Because it is well known (see Kariv and Hakimi 1979) that the PMP is computationally difficult (NP-hard), all the research applying the PMP to cell formation has been concentrating on different types of heuristic solutions. Hence, two types of errors have been incorporated in the solution procedure for CF problems based on the PMP. First, the *modeling error* is caused by an approximation that instead of the intended, and sometimes only, verbally formulated objective function of the CF uses the objective function of the  $p$ -median problem.

That is, solutions that are optimal to the created model may be suboptimal to the original CF problem. Second, heuristic methods for solving the PMP itself give rise to the *algorithmic error*—heuristics do not guarantee optimality of the solutions and often produce suboptimal ones.

In this paper, we adjust a recently suggested compact PMP formulation by AlBdaiwi et al. (2011) (of which the compactness is evaluated by Goldengorin and Krushinsky 2011) to the needs of cell formation problems and show that even large-size CF instances may be solved to optimality in the framework of the PMP by using a general purpose software, such as CPLEX, Xpress, etc., within one second. Moreover, by means of a computational study we show that the modeling error introduced by a PMP-based model is negligible for instances that may be realized in a real-world industrial system. Finally, we show that our approach outperforms several contemporary ones in terms of widely used performance measures.

This paper is organized as follows. The next section describes the  $p$ -median approach to the cell formation problem, including a general formulation of the PMP, its interpretation in terms of cell formation, our efficient MILP formulation, and an analysis of the modeling error

and computational efficiency. Section 3 considers alternate ways of extending the model with additional constraints in order to illustrate its practical applicability. In §4 we provide results of computational experiments with the largest instances used in recent papers (Won and Lee 2004, Yang and Yang 2008, Ahi et al. 2009, Bhatnagar and Saddikuti 2010). Finally, §5 summarizes the paper and outlines possible directions for future research.

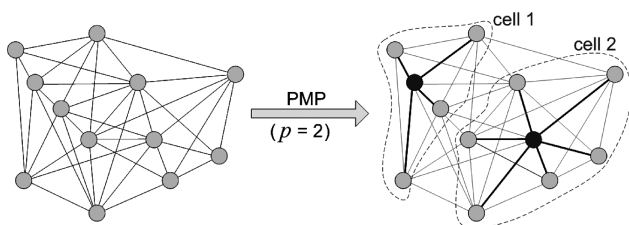
## 2. The $p$ -Median Approach to Cell Formation

The  $p$ -median problem has been applied to cell formation in group technology by a number of researchers (Won and Lee 2004, Deutsch et al. 1998, and references within). However, to the best of our knowledge, in all CF-related papers the PMP is solved by some heuristic method. At the same time, there exist efficient formulations for the  $p$ -median problem (the most recent one derived in Elloumi 2010) that allow solving medium and large-size instances to optimality. In this paper we utilize an innovative model for the PMP that represents an instance in an even more compact way, thus leading to a smaller MILP formulation compared to Elloumi (2010). This allows solving large-scale CF problems to optimality within a second.

The PMP is one of well-known minisum location-allocation problems. A detailed introduction to this problem and solution methods appears in Reese (2006) and Mladenovic et al. (2007). For a directed weighted graph  $G = (V, A, C)$  with  $|V|$  vertices, set of arcs  $(i, j) \in A \subseteq V \times V$  and weights (distances, dissimilarities, etc.)  $C = \{c_{ij} : (i, j) \in A\}$ , the PMP consists of determining  $p$  vertices (the *median vertices*,  $1 \leq p \leq |V|$ ) in such a way that the sum of weights of arcs joining any other vertex and one of these  $p$  vertices is minimized (see Figure 1).

In terms of cell formation, vertices represent machines and weights  $c_{ij}$  represent dissimilarities between machines  $i$  and  $j$ . These dissimilarities can be derived from the sets of parts that are being processed by either of the machines (e.g., if two machines process almost the same set of parts, they have a small dissimilarity and are likely to be in the same cell) or from any other desired characteristics (e.g., workers' skill matrix, operational sequences, etc.). Moreover, usually there is no need to invent a dissimilarity

**Figure 1.** The  $p$ -median problem: minimize the total weight of solid edges.



measure because it can be derived from one of the available similarity measures using an expression  $d(i, j) = c - s(i, j)$ , where  $d(\cdot, \cdot)/s(\cdot, \cdot)$  is a dis/similarity measure and  $c$ —some constant large enough to keep all dissimilarities nonnegative. As can be seen from the literature, several similarity measures were proposed and the particular choice can influence results of cell formation. For our experiments we have chosen one of the most widely used—the “commonality score” of Wei and Kern (1989)—and derived our dissimilarity measure as

$$d(i, j) = r \cdot (r - 1) - \sum_{k=1}^r \Gamma(a_{ik}, a_{jk}), \quad i, j \in \{1, \dots, m\}, \quad (1)$$

where

$$\Gamma(a_{ik}, a_{jk}) = \begin{cases} (r - 1), & \text{if } a_{ik} = a_{jk} = 1; \\ 1, & \text{if } a_{ik} = a_{jk} = 0; \\ 0, & \text{if } a_{ik} \neq a_{jk}; \end{cases} \quad (2)$$

$m$  and  $r$  are the numbers of machines and parts, respectively;  $a_{ik}$ —entries of the *machine-part incidence matrix*—an  $m \times r$  Boolean matrix,  $a_{ik} = 1$  only if part  $k$  needs machine  $i$  at some step of its manufacturing process.

Thus, when applied to cell formation, the  $p$ -median problem means finding  $p$  machines that are best representatives (centers) of  $p$  manufacturing cells, so that the sum over all cells of dissimilarities between such a center and all other machines within the cell is minimized. Once  $p$  central machines are found, the cells are produced by assigning each other machine to the central one, minimizing the dissimilarity. Note that the desired number of cells  $p$  is part of the input for the model and should be known beforehand. Otherwise, it is possible to solve the problem for several numbers of cells and pick the best solution.

Further, for the sake of clarity for those familiar with the PMP, we will follow the terminology inherited from location-allocation applications and represent the set of vertices  $V$  as a union of two, possibly intersecting, sets  $I$  and  $J$ , so that  $|I| = m$ ,  $|J| = n$ . We will call the elements of  $I$ —locations—and those of  $J$ —clients. Moreover, we treat weights  $c_{ij}$  as costs of serving client  $j$  ( $j \in J$ ) from location  $i$  ( $i \in I$ ). In terms of cell formation, the set of locations  $I$  contains potential centers of the cells and the set of clients  $J$  contains all machines. Clearly, in the case of cell formation, sets  $I$  and  $J$  coincide because any machine, potentially, can be a center of a cell. This implies that the PMP applied to cell formation has a symmetric costs matrix.

### 2.1. The MBpBM Formulation

Our approach is based on a compact MILP formulation for PMP—the Mixed-Boolean pseudo-Boolean formulation (MBpBM)—discussed in detail in Goldengorin and Krushinsky (2011). The interested reader is referred to this paper, and here we briefly describe the major idea behind the formulation, which is needed for the further analysis.

The MBpBM formulation is derived from the so-called pseudo-Boolean formulation of PMP (see AlBdaiwi et al. 2011, Goldengorin and Krushinsky 2011) that associates with a cost matrix  $C$  a permutation matrix  $\Pi$ , a differences matrix  $\Delta$ , and a vector of Boolean variables  $\mathbf{y} = (y_1, \dots, y_m)$ , reflecting opened ( $y_i = 0$ ) and closed ( $y_i = 1$ ) locations. Each column of  $\Pi$  is a permutation that sorts the entries from the corresponding column of  $C$  in a nondecreasing order; each column of  $\Delta$  contains differences between consecutive sorted entries of  $C$ . It can be shown that the PMP can be expressed in terms of a polynomial in Boolean variables, abbreviated as  $B_{C,p}(\mathbf{y})$ , with only one constraint requiring exactly  $p$  locations to be opened:  $\sum_{i=1}^m y_i = m - p$ . The pseudo-Boolean formulation can be linearized by introducing for each product of  $y$ -variables in  $B_{C,p}(\mathbf{y})$  a nonnegative  $z$ -variable and a constraint reflecting the relation between  $z$ - and  $y$ -variables. The resulting MBpBM formulation can be expressed as follows:

$$f(\mathbf{y}) = \alpha_0 + \sum_{r=1}^m \alpha_r y_r + \sum_{r=m+1}^{|B|} \alpha_r z_r \rightarrow \min$$

s.t.  $\sum_{i=1}^m y_i = m - p,$

$$z_r \geq \sum_{i \in T_r} y_i - |T_r| + 1, \quad r = m + 1, \dots, |B|,$$

$$z_r \geq 0, \quad r = m + 1, \dots, |B|,$$

$$\mathbf{y} \in \{0, 1\}^m,$$

where  $\alpha_r$  are coefficients of  $B_{C,p}(\mathbf{y})$ ,  $|B|$  denotes the number of monomials in  $B_{C,p}(\mathbf{y})$ , and  $T_r$  is the set of variable indices in monomial  $r$ , that is,  $z_r = \prod_{i \in T_r} y_i$ .

The following example demonstrates how our formulation works for a small CF instance.

EXAMPLE 1. Consider an instance of the cell formation problem defined via the following machine-part incidence matrix

		parts				
		1	2	3	4	5
machines	1		1		1	1
	2	1		1		
	3		1		1	
	4	1		1		

with four machines and five parts (zero entries are skipped for better visualization). One can construct the machine-machine dissimilarity matrix  $C$  by applying the defined above dissimilarity measure (1):

$$C = \begin{bmatrix} 6 & 20 & 10 & 20 \\ 20 & 9 & 19 & 9 \\ 10 & 19 & 9 & 19 \\ 20 & 9 & 19 & 9 \end{bmatrix}. \quad (4)$$

For example, the left top entry  $a_{11}$  is obtained in the following way:

$$\begin{aligned} a_{11} &= r(r-1) - \sum_{k=1}^r \Gamma(a_{1k}, a_{1k}) \\ &= 5(5-1) - \Gamma(0, 0) - \Gamma(1, 1) - \Gamma(0, 0) \\ &\quad - \Gamma(1, 1) - \Gamma(1, 1) \\ &= 20 - 1 - 4 - 1 - 4 - 4 = 6. \end{aligned} \quad (5)$$

The possible permutation and differences matrices for the costs matrix (4) are:

$$\Pi = \begin{bmatrix} 1 & 2 & 3 & 2 \\ 3 & 4 & 1 & 4 \\ 2 & 3 & 2 & 3 \\ 4 & 1 & 4 & 1 \end{bmatrix}, \quad \Delta = \begin{bmatrix} 6 & 9 & 9 & 9 \\ 4 & 0 & 1 & 0 \\ 10 & 10 & 10 & 10 \\ 0 & 1 & 0 & 1 \end{bmatrix}. \quad (6)$$

For example, the first column of the costs matrix  $C^1 = (6, 20, 10, 20)^T$  leads to the following column of  $\Pi(1, 3, 2, 4)^T$  because  $6 \leq 10 \leq 20 \leq 20$ ; that is, the first entry is the smallest, then goes the third entry, then the second and the fourth. The first column  $\Delta^1$  of  $\Delta$  is obtained as follows: the first entry is equal to the smallest element in the first column  $C^1$ , the second entry is the difference between the smallest and the second-smallest element in  $C^1$ , etc. That is,  $\Delta^1 = (6, 10 - 6, 20 - 10, 20 - 20)^T = (6, 4, 10, 0)^T$ .

The above  $\Pi$  and  $\Delta$  lead to the following pseudo-Boolean polynomial  $B_C(\mathbf{y})$ :

$$B_C(\mathbf{y}) = 33 + 4y_1 + 1y_3 + 20y_1y_3 + 20y_2y_4 + 2y_2y_3y_4. \quad (7)$$

If one is interested in having two manufacturing cells, then the number of medians  $p$  in the formulation should be set to 2 and the pseudo-Boolean polynomial can be truncated to the degree of  $(m - p) = 2$ :

$$B_{C,p=2}(\mathbf{y}) = 33 + 4y_1 + 1y_3 + 20y_1y_3 + 20y_2y_4. \quad (8)$$

The obtained polynomial has two nonlinear terms that we need to linearize by introducing additional  $z$ -variables:  $z_5 = y_1y_3$  and  $z_6 = y_2y_4$ . Now, our MBpBM formulation allows expressing the given instance of cell formation as the following mixed-integer LP problem:

$$f(\mathbf{y}, \mathbf{z}) = 33 + 4y_1 + 1y_3 + 20z_5 + 20z_6 \rightarrow \min \quad (9)$$

$$y_1 + y_2 + y_3 + y_4 = 2, \quad (10)$$

$$z_5 \geq y_1 + y_3 - 2 + 1, \quad (11)$$

$$z_6 \geq y_2 + y_4 - 2 + 1, \quad (12)$$

$$z_i \geq 0, \quad i = 5, 6, \quad (13)$$

$$y_i \in \{0, 1\}, \quad i = 1, \dots, 4. \quad (14)$$

Its solution  $y = (0, 0, 1, 1)^T$ ,  $z = (0, 0)^T$  leads to the following cells:

		parts				
		2	4	5	1	3
machines	1	1	1	1		
	3	1	1			
	2				1	1
	4				1	1

(15)

## 2.2. Compactness of the MBpBM Formulation

Taking into account that there is a one-to-one correspondence between nonlinear monomials in  $B_{C,p}(\mathbf{y})$  and nonnegative variables and constraints in MBpBM, the properties of  $B_{C,p}$  directly apply to the MBpBM formulation.

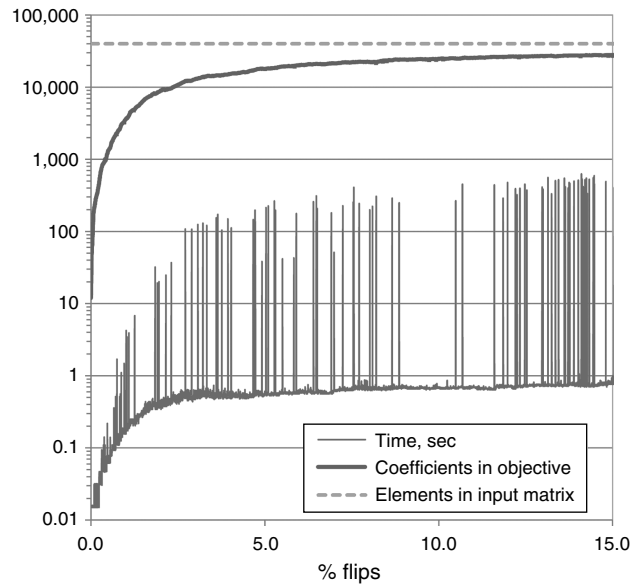
The fundamental property of the pseudo-Boolean formulation is that for real-world instances the number of monomials in  $B_{C,p}(\mathbf{y})$  can be essentially reduced, as compared to the number of entries in the initial costs matrix. In particular, the following three reductions take place:

- only pairwise different elements in each column of the costs matrix play a role;
- all equal column subpermutations in  $\Pi$  contribute to a single monomial in  $B_{C,p}(\mathbf{y})$ ;
- the degree of  $B_{C,p}(\mathbf{y})$  is at most  $m - p$ , i.e., only  $m - p + 1$  smallest different entries in each column of the costs matrix are meaningful (“ $p$ -truncation,” see AlBdaiwi et al. 2011).

The cell formation application supports these reductions. Consider, for example, an instance with  $p$  perfect cells; its machine-part incidence matrix can be transformed into an ideal block-diagonal form with  $p$  blocks. In this case, each column of the corresponding costs matrix for PMP has no more than  $p$  different entries, which is normally much less than  $m - p + 1$ . Next, the number of different subpermutations of each length is also equal to  $p$ . Thus, in case of  $p$  perfect cells the objective has at most  $p \times p$  monomials, irrespective of the number of machines and parts. This results in an MBpBM formulation with at most  $p \times (p - 1)$  nonnegative variables and corresponding constraints, irrespective of the number of machines and parts. Of course, perfect cells are uncommon in practice and the problem becomes larger; however, these considerations demonstrate that the size (and, therefore, complexity) of the model is tightly biased to the complexity of the instance. It should be noticed that the classical formulation of the PMP (which is most widely used, see, e.g., Won and Lee 2004) contains all  $m \times m$  coefficients in the objective function.

To illustrate the computational efficiency, we performed a number of experiments. A  $200 \times 200$  block-diagonal matrix with 5 ideal blocks was generated and then gradually perturbed by adding random flips (change 1 within a diagonal block into 0, or 0 outside a block into 1). For each obtained instance, we estimated the number of coefficients in the objective of the MBpBM formulation and the solution time (using Xpress as a MILP solver). The

**Figure 2.** Performance of a PMP-based model for CF ( $m = 200$ ,  $r = 200$ ,  $p = 5$ ).



size of the instance (200 machines) is intentionally chosen larger than is usual in practice: we could not find instances with more than 50 machines in the literature, although the number of parts does not influence the formulation. This was done in order to show that the performance of our model does not deteriorate with an increase in the instance size. The experimental results are presented in Figure 2, where the numbers of coefficients in the objective function and CPU times are plotted against the amount of flips, expressed as a percentage of the total number of elements in the input matrix. Only the cases with less than 15% of flips are considered, because otherwise the potential inter-cell movement becomes too large and the CF itself does not make sense. As can be seen from the figure, even for the instances with 200 machines the computing times are normally below 1 second, except rare cases (85 out of 6,000) when up to 10 minutes were needed. We believe that these outliers are caused by the MILP solver due to an “imperfect” branching.

Speaking more generally, MBpBM contains all known reductions for PMP not involving presolving the instance, unlike other MILP formulations for PMP. For example, the formulation from Elloumi (2010) does not use the  $p$ -truncation. On the other hand, it is possible to reduce the size of the MBpBM formulation further by involving estimation of lower bounds on the subspaces of feasible solutions (a possible framework is described in Goldengorin and Krushinsky 2011).

## 2.3. A Note on Optimality of PMP-Based Models

Note that the PMP does not explicitly optimize the goal of cell formation. In particular, the modeling error can be explained by the fact that each machine is assigned to a

cell based on the similarity between itself and the median machine, without taking into account other machines from this cell. This means that even having solved the PMP (after finding median machines), there is some space for improvement at a stage of assigning other machines to cells. However, here we analyze the quality of solutions produced by a standard PMP-based model using the dissimilarity measure (1).

Let us first consider the case of a manufacturing system in which  $p$  perfect cells are possible. It is not hard to understand that a PMP model equipped with a reasonable dissimilarity measure (like the one described in (1)) will discover those  $p$  cells, thus producing optimal results. In practice, however, perfect cellular structure is distorted to some extent. If input data are given in a form of a machine-part incidence matrix, then there are two types of distortions: *voids*—zeros in diagonal blocks, and *exceptions*—ones outside the diagonal blocks.

The following propositions provide sufficient conditions for optimality of the obtained solution.

**PROPOSITION 1.** *Suppose a block structure without exceptions exists (only voids are allowed). In this case, a PMP-based model produces an optimal solution if in each cell there is at least one machine that is needed for all parts from the corresponding part family.*

**PROOF OF PROPOSITION 1.** First we prove that only machines needed for all parts in the cells can become medians. Observe that the dissimilarity measure (1) is designed in such a way that for any two machines (rows of the machine-part incidence matrix), each coinciding one weighs more than any number of zeros. This implies that only the machine that is needed for all parts assigned to a cell will “cover” the maximum number of ones and will be selected as a median. As soon as medians are defined, all other machines are uniquely assigned to the cells where they are needed—assumption of the proposition implies that a structure where each machine is needed in exactly one cell is possible. As a result, completely independent cells will be obtained.  $\square$

**PROPOSITION 2.** *Suppose, a solution with a block structure without exceptions (only voids are allowed) is found by a PMP model. If in each cell the median machine has at least one part in common with any other machine in a cell, then this solution is optimal to CF.*

**PROOF OF PROPOSITION 2.** Straightforward, because moving any machine to a different cell will create at least one exception.  $\square$

**PROPOSITION 3.** *Suppose a solution with a block structure without voids is found (only exceptions are allowed). If the number of exceptions in each row is strictly less than the number of within-block ones in this row, then the solution is optimal.*

**PROOF OF PROPOSITION 3.** Absence of voids in the blocks guarantees that the assignment of machines (rows) to cells (blocks) is not sensitive to a particular choice of medians. At the same time, a limited number of exceptions induced by any machine guarantees that its current position is optimal, irrespective of the configuration of other blocks. This is due to the fact that moving a machine to the other cell will reduce the number of matching ones, and this cannot be compensated with any increase in the number of matching zeros, due to the used dissimilarity measure (1).  $\square$

Proposition 3 can be generalized to allow for both voids and exceptions.

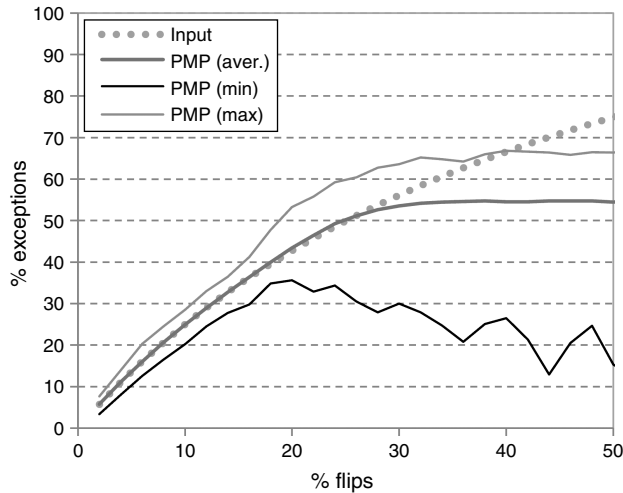
**PROPOSITION 4.** *If there exists an optimal solution to the CF problem satisfying the following requirement, then it will be found by a PMP-based model: for any two machines (rows)  $i$  and  $j$  belonging to the same cell (block)  $k$  the total number of voids in rows  $i$  and  $j$  is strictly less than the difference between the number of parts (columns) assigned to cell  $k$  and the number of exceptions in either of the rows  $i$  and  $j$ . The inverse is also true: if such a solution is found, then it is optimal.*

**PROOF OF PROPOSITION 4.** The condition ensures that any machine (row) has more matching ones with any other machine from the same cell (by the pigeonhole principle) than with a machine from another cell. This guarantees that the assignment of machines (rows) to cells (blocks) is not sensitive to a particular choice of medians. The rest of the reasoning is similar to the proof of Proposition 3.  $\square$

As can be observed from Propositions 1–4, presence of dense blocks is critical for an optimality unless exceptional elements can be avoided. These propositions assume some properties of the optimal solution; thus, they can only be used for a posterior assessment of optimality. However, as our numerical experiments show, the solution time for our model is very small, and it is reasonable to solve the problem and then check the optimality of the obtained solution.

As the conditions of Propositions 1–4 are not always met, we performed an experimental study on the possible modeling error introduced by the PMP models for CF. We generated input matrices with an ideal block-diagonal structure and then gradually destroyed it by adding (unbiased) random flips (change 1 within a diagonal block into 0, or 0 outside a block into 1). For each instance we compared the performance of the original configuration of the cells and the one discovered by our PMP-based model in terms of the number of exceptions (expressed as a percentage of the total number of ones in the matrix). The latter quantity is exactly the volume of intercell movement. The typical behavior of a PMP-based model is presented in Figure 3, where the number of flips is expressed as a percentage of the total number of elements  $m \times r$  in the input matrix and each data point is averaged for about 1,000 trials. As the figure shows, the average error is quite low and

**Figure 3.** Solution quality of a PMP-based model for CF ( $m = 25$ ,  $r = 50$ ,  $p = 4$ ; cell sizes vary from  $4 \times 9$  to  $8 \times 15$ ).



does not exceed 1%. The maximum error in our experiments was also quite limited and did not exceed 10%. Clearly, as the number of flips gets larger, the initial configuration of cells is no more optimal and becomes dominated. It can be easily checked that as the amount of flips approaches 50%, the matrix approaches a completely random one (i.e., each element is 1 with probability 0.5). It is also important to understand that in this case the CF problem itself does not make sense because the underlying system does not possess a cellular structure and cannot be decomposed in a reasonable way. In fact, cellular decomposition of the manufacturing system makes sense in practice only if the resulting volume of intercell movement (exceptions in the block-diagonalized matrix) is below 10%–15%; in these cases, the maximum modeling error in our experiments does not exceed 4%, whereas the average error is of the order  $10^{-3}\%$ .

### 3. Possible Extensions of the Model

In this section we would like to discuss the possibilities of introducing additional real-life factors and constraints into the model. Thus, we are not interested here in describing all constraints that can be incorporated, but rather in demonstrating the possibility of extending the model appropriately.

Clearly, our model has three places in which additional factors can be incorporated:

- dissimilarity coefficients
- objective function (structure)
- constraints

The use of dissimilarity coefficients can be illustrated, for example, as follows. The availability of skills in a manufacturing system can be represented by a machine-worker skills matrix, in a way very similar to the input for machine-part grouping. This means that any available

machine-machine (dis)similarity measure can be applied to this skills matrix. Plugged into a similarity-based cell formation approach, this measure minimizes the number of workers that can operate a machine outside of their cell, or, equivalently, maximizes the number of machines that each worker can operate within his cell. Similarities based on either of these data can be combined in a number of ways (e.g., linearly or multiplicatively). The case of a linear combination with equal weighting coefficients is equivalent to having one aggregated incidence matrix where each column corresponds either to a part or to a worker. It should be mentioned that the same approach is used in Bhatnagar and Saddikuti (2010) for what they call a concurrent model. In their paper it is also demonstrated that such an approach gives better results than two-stage procedures that make cells and assign workers consecutively.

The objective function can be extended, for example, by penalizing assignments of some machines to the same cell. In this way an issue of equivalent machines (the ones with similar functionality) can be resolved by forcing them to different cells. Another example is the use of manufacturing sequences: terms accounting for multiple transits of a part between the cells can be added to the objective.

Finally, a wide variety of linear constraints can be included. These range from simple variable fixing constraints, to capacity, workload balancing, etc. For example, simply by fixing some  $z$ -variables one can force or prohibit the assignment of some machines to the same cells—this could be necessary because of safety, engineering, or managerial considerations. Although it may be complex to linearize some constraints, those normally occurring in CF are capacity-like, and therefore linear.

We would like to conclude the section by stating that the reductions making our model efficient are based exclusively on the properties of the underlying clustering model and assume nothing about its further extension. This implies that any additional constraints expressed in a linear form can be added to our compact formulation.

### 4. Experimental Results

The aim of our numerical experiments is twofold. First, we would like to show that the model based on PMP produces high-quality cells and in most cases outperforms other contemporary approaches, thus making their use questionable. Second, by showing that computation times are negligibly small, we argue the use of heuristics for solving PMP itself.

Up to this point, one basic notion remained undefined in this paper—the quality measure of the obtained decomposition into cells. We used two most widely used measures so as to ensure consistent comparison of results. The first one, the group capability index (GCI) proposed by Hsu (1990), can be calculated as follows:

$$\text{GCI} = 1 - \frac{\text{number of exceptional elements}}{\text{total number of ones}} \times 100\%, \quad (16)$$

where *exceptional elements* are the nonzero entries of the block-diagonalized machine-parts coincidence matrix that lie outside of the blocks and the *total number of ones* is the total number of nonzero entries in the machine-parts incidence matrix. It should be mentioned that this measure does not account for zeros inside the blocks, i.e., it does not take into account density of intracell flows. The second quality measure, group efficiency ( $\eta$ ), was proposed by Chandrasekharan and Rajagopalan (1986) and is a weighted sum of two factors  $\eta_1$  and  $\eta_2$ :

$$\eta = \omega\eta_1 + (1 - \omega)\eta_2 \times 100\%. \quad (17)$$

In turn,  $\eta_1$  and  $\eta_2$  are expressed as:

$$\eta_1 = \frac{o - e}{o - e + v},$$

$$\eta_2 = \frac{mr - o - v}{mr - o - v + e},$$

where  $m$ —number of machines,  $r$ —number of parts,  $o$ —number of ones in the part-machine matrix,  $e$ —number of exceptional elements,  $v$ —number of zeros in diagonal blocks. The weighting factor  $\omega$  is usually set to 0.5, and we used this value.

For the instance considered above (15), these performance measures have the following values: GCI = 100%,

$$\eta = 0.5 \cdot \left( \frac{9 - 0}{9 - 0 + 1} + \frac{20 - 9 - 1}{20 - 9 - 1 + 0} \right) \times 100\% = 95\%.$$

It should be mentioned that the sum of voids and exceptions ( $v + e$ ) sometimes is used as a performance measure (see, e.g., Bhatnagar and Saddikuti 2010).

Taking into account the aim of our experiments, we compared our results with those reported in four recent papers and an earlier paper by Chen and Heragu (1999). The main focus was on the largest instances. The first paper is by Won and Lee (2004) and, similar to our study, uses a  $p$ -median approach but solves PMP with a heuristic procedure. They use the similarity measure of Wei and Kern (1989) and GCI (16) as a quality measure. We were not able to derive the value of  $\eta$  because solutions are not provided in their paper. The second paper by Yang and Yang (2008) applies the ART1 neural network to cell formation, thus using a completely different approach. The authors used  $\eta$ -measure (17) to estimate solution quality and included solutions (block-diagonalized matrices) in their paper, making it possible for us to compute GCI and to fill in the gaps in the following Table 1, which summarizes results of our comparative experiments. The third paper is by Ahi et al. (2009) and demonstrates an application of a decision-making technique (TOPSIS) to the cell formation problem. The authors report values of group efficiency  $\eta$ , and we derived the values of GCI from their solutions.

Table 1 contains data on computational experiments with instances used in the three papers mentioned above

**Table 1.** Experimental comparison with Won and Lee (2004), Yang and Yang (2008), and Ahi et al. (2009).

Source	$m \times r$	$p$	GCI	GCI <sub>our</sub>	$\eta$	$\eta_{our}$	
Won and Lee (2004)	$30 \times 41$	3	92.2	95.3		59.38	
		4	93.0	93.0		64.39	
		5	91.4	91.4		72.14	
		6	89.8	90.6		75.25	
		7	81.3	89.8		77.93	
		$30 \times 50$	3	77.2	77.3		59.53
			4	74.9	76.1		62.14
	$30 \times 90$		3	79.9	77.5		61.00
			2	79.5	93.6		55.61
	$40 \times 100$	3	93.1	91.5		59.59	
		4	89.8	88.8		63.84	
		5	89.3	87.4		69.33	
		6	89.3	88.1		75.77	
		7	87.6	88.6		81.38	
		8	85.5	89.1		85.66	
		$50 \times 150$	2	96.5	96.5		57.49
	3		86.4	90.1		62.63	
	4		88.4	92.7		69.05	
	5		89.7	91.5		76.44	
	6		87.3	93.1		81.89	
	Yang and Yang (2008)		$28 \times 35$	6	73.7	73.7	90.68
$46 \times 105$		7	84.1	84.9	87.54	87.57	
Ahi et al. (2009)	$8 \times 20$	3	83.6	83.6	92.11	98.08	
	$12 \times 19$	4	66.2	66.2	80.10	77.09	
	$20 \times 20$	6	67.1	82.3	87.89	90.11	
	$18 \times 35$	4	77.2	77.2	74.10	81.26	
	$25 \times 40$	7	61.5	76.2	68.02	85.43	
	$20 \times 51$	6	67.8	77.2	82.62	82.07	

(Won and Lee 2004, Yang and Yang 2008, Ahi et al. 2009). Column “source” indicates the source of the cell formation instance and of the performance data. The next two columns contain information on the size of input, such as the number of machines  $m$ , the number of parts  $r$ , and the number of cells to be made  $p$ . The last four columns indicate the quality of solutions (in terms of GCI and  $\eta$ ) reported in the discussed papers and obtained by us, correspondingly.

As can be seen from Table 1, in most of the considered cases our results outperform those reported in literature. On the other hand, there exist scarce instances for which our model is dominated by other heuristics. This can be explained by the fact that even though we solve PMP to optimality, the  $p$ -median problem itself is not explicitly an exact model to optimize any of the quality measures of cell decomposition used above (their appropriateness can also be debated). Consequently, any model based on the  $p$ -median problem is of a heuristic nature. However, unlike most of the other heuristics, it grasps the clustering nature of cell formation and presents a flexible framework by allowing additional constraints reflecting real-world manufacturing systems to be introduced. Such flexibility is inherent, in particular, to mathematical

programming approaches, but in contrast to them, for the PMP we have found an efficient formulation (see §2.1).

The most recent paper considered in our computational experiments is by Bhatnagar and Saddikuti (2010). It uses a model that is very similar to the  $p$ -median problem but differs in the following detail: a restriction specifying the number of cells is replaced by a constraint ensuring that each cell has at least two machines. However, in our opinion, this model has a potential drawback because it tends to split “reasonable” cells as can be seen from its objective function (taking into account that for similarities holds  $s(i, i) \geq s(i, j)$  for any two machines  $i, j$ ). We implemented the models for machine cell formation and part assignment from Bhatnagar and Saddikuti (2010) in Xpress and performed a number of experiments with the largest instances available in literature. As in the previous cases, we used only machine-part incidence matrices as an input, and the (dis)similarity measure of Wei and Kern’s (1989). Taking into account that the model from Bhatnagar and Saddikuti (2010) automatically defines the best number of cells, we had to solve our PMP-based model for all possible values of  $p$  and pick the best results.

Finally, we compared performance of our model and the one from Chen and Heragu (1999), which we implemented in Xpress. Because this model, like ours, does not define the optimal number of cells, we tried to solve it for all possible values of  $p$ . However, this was not always possible

due to the complexity of the model. We limited the CPU time of the model by 10 hours, and provided the best results that we could obtain. These results can be suboptimal for two reasons: either (i) a solution process was interrupted after the time limit was reached, or (ii) a solution is optimal for a particular value of  $p$ , but is not globally optimal (for other values of  $p$  case (i) takes place and gives even worse solutions). Note that the formulation does not use a dissimilarity measure and deals directly with machine-part relations. This makes the applicability of the model questionable, because in real systems the number of parts can be estimated in thousands, leading to a huge formulation even though the number of machines is small. For example, an instance with 4,415 parts was considered by Park and Suresh (2003), and we experienced even larger ones in the industry.

The results for our model and the ones from Bhatnagar and Saddikuti (2010) and Chen and Heragu (1999) are summarized in Table 2, where the first column enumerates the test instances, the second one refers to the original source of the instance, and the next column shows the number of machines and parts. The following six columns report the quality of solutions obtained by the three models. The last column indicates the time spent (in seconds) by the model from Chen and Heragu (1999). As can be seen from Table 2, our model considerably outperforms the model from Bhatnagar and Saddikuti (2010).

**Table 2.** Experimental comparison of our model and those from Bhatnagar and Saddikuti (2010) and Chen and Heragu (1999).

No.	Source	$m \times r$	$(e + v)$			$\eta(\%)$			Time (s)
			[BS10]	[CH99]	Our	[BS10]	[CH99]	Our	[CH99]
1	Sandbothe (1998)*	20 × 10	16	9	11	95.40	94.29	95.93	2
2	Ahi et al. (2009)	20 × 20	34	26	26	92.62	90.70	93.85	2,385
3	Mosier and Taube (1985)*	20 × 20	79	74	77	85.63	79.51	88.71	36,000
4	Boe and Cheng (1991)*	20 × 35	87	77	83	88.31	84.44	88.05	24,724
5	Carrie (1973)*	20 × 35	46	40	41	90.76	88.93	95.64	110
6	Ahi et al. (2009)	20 × 51	111	96	83	87.86	83.18	94.11	36,000
7	Chandrasekharan and Rajagopalan (1989)*	24 × 40	20	0	0	98.82	100.00	100.00	2
8	Chandrasekharan and Rajagopalan (1989)*	24 × 40	37	21	21	95.33	95.20	97.48	1,363
9	Chandrasekharan and Rajagopalan (1989)*	24 × 40	55	40	39	93.78	91.16	96.36	29,009
10	Chandrasekharan and Rajagopalan (1989)*	24 × 40	86	122	81	87.92	74.38	94.32	14,890
11	Chandrasekharan and Rajagopalan (1989)*	24 × 40	96	112	89	84.95	77.68	94.21	10,968
12	Chandrasekharan and Rajagopalan (1989)*	24 × 40	94	118	89	85.06	75.29	92.32	16,906
13	Nair and Narendran (1996)*	24 × 40	40	194	25	96.44	69.90	97.39	36,000
14	Nair and Narendran (1996)*	24 × 40	39	27	26	92.35	92.27	95.74	3,575
15	Nair and Narendran (1996)*	24 × 40	60	50	50	93.25	90.56	95.70	36,000
16	Nair and Narendran (1996)*	24 × 40	59	109	50	91.11	78.08	96.40	36,000
17	Ahi et al. (2009)	25 × 40	59	63	56	91.09	86.00	95.52	36,000
18	Yang and Yang (2008)	28 × 35	108	72	71	93.43	91.21	93.82	36,000
19	Kumar and Vanelli (1987)*	30 × 41	63	61	54	90.66	86.78	97.22	16,967
20	Stanfel (1985)*	30 × 50	99	115	93	88.17	81.58	96.48	36,000
21	King and Nakornchai (1982)*	30 × 90	228	202	206	83.18	83.25	94.62	36,000
22	Chandrasekharan and Rajagopalan (1987)*	40 × 100	136	72	72	94.75	95.91	95.91	36,000
23	Yang and Yang (2008)	46 × 105	376	268	271	90.98	87.12	95.20	36,000
24	Zolfaghari and Liang (1997)*	50 × 150	544	502	470	93.05	82.00	92.92	36,000

Notes. [BS10]—results for the model from Bhatnagar and Saddikuti (2010). [CH99]—results for the model from Chen and Heragu (1999).

\*A reference to the original source of the instance can be found in Bhatnagar and Saddikuti (2010).



Also, our computational experiments show that in terms of the group efficiency criterion (17), the PMP-based model essentially outperforms Chen and Heragu's results but is slightly dominated in 6 out of 24 benchmark instances in terms of the less widely used ( $e + v$ ) criterion. This can be explained by the fact that their model explicitly optimizes the latter criterion. Our and Chen's and Heragu's (1999) average CPU times are within 1 sec and many hours, respectively, and this distinction cannot be attributed to the operational system, or to the hardware used in experiments. The difference in the CPU times can be explained by the adaptability of our model to the input data. Consider, for example, instance 7 from Table 2 that has a perfect cellular structure with 7 cells. For this instance, our MBpBM formulation has 42 variables, 19 constraints, and 19 coefficients in the objective, whereas for the model from Chen and Heragu (1999) these numbers are 7,168, 6,851, and 20,224. At the same time, for instances with perfect cells our model provides provably optimal solutions! This can be verified using any of Propositions 1–4.

We have also considered a few other models, such as the one from Doulabi et al. (2009); these all were dominated by our model.

Concerning the solution times of our PMP-based model, each of the considered instances was solved within one second on a PC with 2.3 GHz Intel processor, 2 GB RAM, and Xpress as a MILP solver. In our opinion, even if some heuristic can work faster, the difference in computing times is negligibly small.

## 5. Summary and Future Research

As can be seen from the available literature, the models for cell formation tend to become more and more complicated. Such complication has two negative side effects. First of all, a sophisticated structure of the model usually prohibits its extension to additional factors and/or constraints taking place in real manufacturing systems. Secondly, a complicated model that was designed in order to improve the quality of the obtained solutions usually raises a problem of computational intractability. This forces the use of heuristics for solving not the initial cell formation problem but an approximating model of it. Suboptimality of these heuristics can overwhelm the advantages of the model, making its use doubtful.

In this paper we showed that these negative side effects can be avoided by efficiently reformulating the cell formation problem in terms of a compact representation of the  $p$ -median problem (PMP). Our reformulation is flexible enough to accept additional real-life constraints, such as capacities, operational sequences, etc. At the same time, the computational experiments show that our model is computationally efficient and the corresponding problem can be solved to optimality within one second on a standard PC by means of general-purpose software, such as CPLEX or Xpress. A computational study shows that in most cases our

cell formation model provides better solutions than other contemporary approaches (in terms of widely used quality measures) while having shorter CPU times. Finally, by means of computational experiments we demonstrated that the modeling error of a PMP-based model is quite limited with an average of 1%, and solution times stay within one second in 99% of the cases, even for instances with 200 machines, i.e., larger than those occurring in practice.

Taking into account that the current trend is towards introducing additional real-world factors into CF models, a possible future research direction is to incorporate additional constraints into our model, such as an availability of several equivalent machines, alternative operational sequences, setup and processing times, etc. Because our MBpBM formulation is optimal in the number of coefficients in the objective function and the number of linear constraints, insertion of new (linear) constraints will preserve its tractability and will make it possible to create a flexible and efficient model for cell formation based on the  $p$ -median problem, in our opinion. The issue of efficiency (low computing times) is becoming important from the perspective of virtual cell manufacturing (VCM, see Nomden et al. 2006)—a paradigm that becomes more and more promising nowadays. Slomp et al. (2005) suggest that the VCM concept needs a periodic (e.g., weekly) solution for the CF problem. At the same time, our computational results show that in the case of an uncapacitated functional grouping, our fast model is a competitive candidate for the creation of virtual cells.

To summarize, all ideas and attempts to extend the decision making for cell formation in group technology based on the classical  $p$ -median model might be revised and essentially improved by using the MBpBM reformulation and adding practically motivated additional constraints reflecting the specific manufacturing environment. Finally, we would like to stress the importance of the model choice and to conclude by asserting that the above considerations on the problem complexity and errors involved can also be valid for other applied operations research problems (especially for those that can be modeled as variations of the PMP).

## Acknowledgments

The authors of this paper have benefited in many respects by means of the reviewing procedure feedback from both reviewers and the corresponding area editor, who has provided essential guidance to revise and improve the quality of this paper. Boris Goldengorin was supported by the grant of National Research University Higher School of Economics within the project no. 11-04-0008 "Calculus of tolerances in combinatorial optimization: Theory and algorithms," 2011 and the grant of Russian Federation Government 11.G34.31.0057.

## References

- Ahi A, Aryanezhad MB, Ashtiani B, Makui A (2009) A novel approach to determine cell formation, intracellular machine layout and cell layout in the CMS problem based on TOPSIS method. *Comput. Oper. Res.* 36(5):1478–1496.

- AlBdaiwi B, Ghosh D, Goldengorin B (2011) Data aggregation for  $p$ -median problems. *J. Combin. Optim.* 21(3):348–363.
- Ashayeri J, Heuts R, Tammel B (2005) A modified simple heuristic for the  $p$ -median problem, with facilities design applications. *Robot. Comput.-Integrated Manufacturing* 21(4–5):451–464.
- Bhatnagar R, Saddikuti V (2010) Models for cellular manufacturing systems design: Matching processing requirements and operator capabilities. *J. Oper. Res. Soc.* 61(5):827–839.
- Chandrasekharan MP, Rajagopalan R (1986) An ideal seed nonhierarchical clustering algorithm for cellular manufacturing. *Internat. J. Production Res.* 24(2):451–463.
- Chen J-S, Heragu SS (1999) Stepwise decomposition approaches for large scale cell formation problems. *Eur. J. Oper. Res.* 113(1):64–79.
- Deutsch SJ, Freeman SF, Helander M (1998) Manufacturing cell formation using an improved  $p$ -median model. *Comput. Indust. Engrg.* 34(1):135–146.
- Doulabi SHH, Hojabri H, Seyed-Alagheband S-A, Jaafari AA, Davoudpour H (2009) Two-phase approach for solving cell-formation problem in cell manufacturing. *Proc. World Congress Engrg. Comput. Sci. WCECS 2009*, Vol. 2 (Newswood Limited, Hong Kong), 1226–1231.
- Elloumi S (2010) A tighter formulation of the  $p$ -median problem. *J. Combin. Optim.* 19(1):69–83.
- Goldengorin B, Krushinsky D (2011) Complexity evaluation of benchmark instances for the  $p$ -median problem. *Math. Comput. Modelling* 53(9–10):1719–1736.
- Hsu CP (1990) Similarity coefficient approaches to machine-component cell formation in cellular manufacturing. Ph.D. thesis, Department of Industrial and Systems Engineering, University of Wisconsin, Milwaukee.
- Kariv O, Hakimi SL (1979) An algorithmic approach to network location problems. II: The  $p$ -medians. *SIAM J. Appl. Math.* 37(3):539–560.
- Kusiak A, Chow WS (1987) Efficient solving of the group technology problem. *J. Manufacturing Systems* 6(2):117–124.
- Mladenovic N, Brimberg J, Hansen P, Moreno-Perez JA (2007) The  $p$ -median problem: A survey of metaheuristic approaches. *Eur. J. Oper. Res.* 179(3):927–939.
- Nomden G, Slomp J, Suresh NC (2006) Virtual manufacturing cells: A taxonomy of past research and identification of future research issues. *Internat. J. Flexible Manufacturing Systems* 17(2):71–92.
- Park S, Suresh NC (2003) Performance of fuzzy ART neural network and hierarchical clustering for part-machine grouping based on operation sequences. *Internat. J. Production Res.* 41(14):3185–3216.
- Reese J (2006) Solution methods for the  $p$ -median problem: An annotated bibliography. *Networks* 48(3):125–142.
- Selim HM, Askin RG, Vakharia AJ (1998) Cell formation in group technology: Review, evaluation and directions for future research. *Comput. Indust. Engrg.* 34(1):3–20.
- Slomp J, Chowdary BV, Suresh NC (2005) Design of virtual manufacturing cells: A mathematical programming approach. *Robot. Comput. Integrated Manufacturing* 21(3):273–288.
- Wang J, Roze C (1997) Formation of machine cells and part families: A modified  $p$ -median model and a comparative study. *Internat. J. Production Res.* 35(5):1259–1286.
- Wei JC, Kern GM (1989) Commonality analysis. A linear cell clustering algorithm for group technology. *Internat. J. Production Res.* 27(12):2053–2062.
- Won Y, Currie KR (2006) An effective  $p$ -median model considering production factors in machine cell/part family formation. *J. Manufacturing Systems* 25(1):58–64.
- Won Y, Lee KC (2004) Modified  $p$ -median approach for efficient GT cell formation. *Comput. Indust. Engrg.* 46(3):495–510.
- Yang M-S, Yang J-H (2008) Machine-part cell formation in group technology using a modified ART1 method. *Eur. J. Oper. Res.* 188(1):140–152.
- Yin Y, Yasuda K (2006) Similarity coefficient methods applied to the cell formation problem: A taxonomy and review. *Internat. J. Production Econom.* 101(2):329–352.

**Boris Goldengorin** is a professor in the Department of Higher Mathematics and Deputy Head in LATNA at the National Research University Higher School of Economics, Moscow, and professor in the Department of Mathematics and Informatics at the Khmel'nitsky University of Economics and Business. He has published widely in data correcting, tolerances, and patterns-based algorithms for solving routing, location, scheduling, clustering, and cell formation problems.

**Dmitry Krushinsky** is a postdoc researcher at the Faculty of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, the Netherlands. During the preparation of this paper, he was a Ph.D. student at the Department of Operations, Faculty of Economics and Business, University of Groningen, the Netherlands, where he obtained a Ph.D. degree in economics and business. His research interests lie mainly within the area of applied operations research and mathematical modeling of complex systems.

**Jannes Slomp** is a professor at the Faculty of Engineering at the HAN University of Applied Sciences, the Netherlands. His research interests include flexible manufacturing systems, cellular manufacturing, lean manufacturing, and modular product development.