Pattern-Based Heuristic for the Cell Formation Problem in Group Technology

Mikhail Batsyn, Ilya Bychkov, Boris Goldengorin, Panos Pardalos, and Pavel Sukhov

Abstract In this chapter we introduce a new pattern-based approach within the linear assignment model with the purpose to design heuristics for a combinatorial optimization problem (COP). We assume that the COP has an additive (separable) objective function and the structure of a feasible (optimal) solution to the COP is predefined by a collection of cells (positions) in an input file. We define a *pattern* as a collection of positions in an instance problem represented by its input file (matrix). We illustrate the notion of pattern by means of some well-known problems in COP, among them are the linear ordering problem (LOP) and cell formation problem (CFP), just to mention a couple. The CFP is defined on a Boolean input matrix, the rows of which represent machines and columns – parts. The CFP consists in finding three optimal objects: a block-diagonal collection of rectangles, a row (machines) permutation, and a column (parts) permutation such that the grouping efficacy is maximized. The suggested heuristic combines two procedures: the pattern-based procedure to build an initial solution and an improvement procedure to obtain a final solution with high grouping efficacy for the CFP. Our computational experiments with the most popular set of 35 benchmark instances show that our heuristic outperforms all well-known heuristics and returns either the best known or improved solutions to the CFP.

Keywords Cell formation problem • Group technology • Heuristic

M. Batsyn (✉) • I. Bychkov • B. Goldengorin • P. Pardalos • P. Sukhov
Laboratory of Algorithms and Technologies for Network Analysis The National Research University Higher School of Economics, 136, Rodionova Street, Nizhny Novgorod, Russian Federation, 603093, Russia
e-mail: mbatsyn@hse.ru; il.bychkov@gmail.com; bgoldengorin@hse.ru; pardalos@ufl.edu; pavelandreevith@rambler.ru

P. Pardalos
Center of Applied Optimization, University of Florida, 401 Weil Hall, Gainesville, FL 32611-6595, P.O. Box 116595, USA
e-mail: pardalos@ufl.edu

B. Goldengorin et al. (eds.), *Models, Algorithms, and Technologies for Network Analysis*, Springer Proceedings in Mathematics & Statistics 32, DOI 10.1007/978-1-4614-5574-5_2,

1 Introduction

The cell formation problem (CFP) in group technology was formulated more than 50 years ago and has gained a lot of attention in the industrial engineering literature (see, e.g., Mitrofanov [33, 34], Burbidge [6, 7]).

Ballakur and Steudel [3] have shown that the CFP with different objective functions is an NP-complete (hard) problem. That is why there is a great number of heuristics for solving CFP. In general case, NP-hardness does not imply that for some specific objective functions and practically motivated sizes of the CFP this problem cannot be solved to optimality. The most recent examples of exact algorithms are presented in Krushinsky and Goldengorin [22] for solving the CFP by means of MINpCUT model and Goldengorin et al. [16] by means of the p-median model.

A large class of heuristics for solving the CFP is represented by different clustering techniques. Array-based clustering methods find row and column permutations of the machine-part matrix in order to form a block-diagonal structure. These methods include bond energy algorithm (BEA) of McCormick et al. [32], rank order clustering (ROC) algorithm by King [20], its improved version ROC2 by King and Nakornchai [21], direct clustering algorithm (DCA) of Chan and Milner [9], modified rank order clustering (MODROC) algorithm by Chandrasekaran and Rajagopalan [11], and close neighbor algorithm (CAN) by Boe and Cheng [5]. Hierarchical clustering methods at first form several big cells and then divide each cell into smaller ones and so on gradually improving the value of the objective function. The most well-known methods are single linkage [31], average linkage [43], and complete linkage [35] algorithms. Nonhierarchical clustering methods are iterative approaches which start from some initial partition and improve it iteratively. The two most successful heuristics are GRAFICS heuristic by Srinivasan and Narendran [46] and ZODIAC heuristic by Chandrasekharan and Rajagopalan [12]. There are many heuristics based on the graph partitioning approach to the CFP (see, e.g., Rajagopalan and Batra [41], Askin and Chiu [1]), the minimum spanning tree (see Ng [38, 39]), and mathematical programming approach (see, e.g., Kusiak [26], Shtub [44], Won and Lee [50], Krushinsky and Goldengorin [22], Goldengorin et al. [16]). Also meta-heuristics have been applied to the CFP (see, e.g., Goncalves and Resende [17], Wu et al. [51], Xambre and Vilarinho [52], Lei and Wu [29], Liang and Zolfaghari [30], Yang and Yang [53]).

The purpose of this chapter is twofold. First we coin the notion of a *pattern* within the linear assignment model and second we will apply this notion to design a new pattern-based heuristic for solving the CFP. Our chapter is organized as follows.

In the next section we introduce the notion of pattern and illustrate it by means of combinatorial optimization problems (COPs), for example, the CFP. In Sect. 3 we present our heuristic combining the pattern-based and improvement procedures. In Sect. 4 we report our promising computational results and Sect. 5 concludes the chapter with a summary and future research directions.

2 Patterns and the CFP

In this section we introduce the notion of pattern and illustrate it by means of the linear ordering problem (LOP) and the CFP including different objective functions used in the literature.

2.1 Patterns

We define a *pattern* as a specific collection of cells in the given input data (matrix) reflecting the structure of a feasible (optimal) solution to the original combinatorial optimization problem (COP). The following COPs are considered as examples defined on an input matrix. We are given a matrix and a pattern (a collection of positions in the matrix) defined on this matrix. The COP objective function is to find optimal row and column permutations which minimize (maximize) the sum of elements appearing in the pattern after applying these permutations to the matrix. Many COPs such as the assignment problem (AP), LOP or triangulation problem, traveling salesman problem (TSP), and maximum clique problem (MCP) can be formulated within the pattern-based AP model. Examples of patterns for different COPs are provided below.

The first example is the AP formulated on a square input matrix of order $n = 5$ (see Table 1) as follows. An arbitrary single element $a(i, j)$ of the input matrix is called an *assignment of row i to column j* with its value $a(i, j)$. The AP is the problem of finding a one-to-one mapping of rows to columns by means of entries $a(i, j)$ such that the total sum of all n entries is minimized. The AP pattern is defined as any collection of exactly n cells (positions) located in pairwise distinct rows and columns, i.e., each row (column) contains exactly one cell (see Table 2). For the sake of simplicity we have chosen n cells located at the main diagonal, namely $\mathscr{P}(AP) = \{(1,1),\ldots, (i,i),\ldots,(n,n)\}$ representing an AP pattern. The AP is the

Table 1 Original matrix

	c_1	c_2	c_3	c_4	c_5
r_1	2	8	4	5	3
r_2	4	9	6	7	1
r_3	3	4	2	0	9
r_4	1	0	9	8	4
r_5	8	3	7	6	5

Table 2 AP pattern

	1	2	3	4	5
1	■				
2		■			
3			■		
4				■	
5					■

Table 3 Entries r_1 located at the first row

	1	2	3	4	5	
r_1	2	8	4	5	3	1
						2
						3
						4
						5

Table 4 Entries r_1 located at the second row

	1	2	3	4	5	
						1
r_1	2	8	4	5	3	2
						3
						4
						5

Table 5 All contributions of r_1 to the objective function of AP

	1	2	3	4	5	
r_1	2	8	4	5	3	1
r_1	2	8	4	5	3	2
r_1	2	8	4	5	3	3
r_1	2	8	4	5	3	4
r_1	2	8	4	5	3	5

problem of finding a permutation of rows such that the total sum of all entries appearing within all cells of $\mathscr{P}(AP)$ pattern is minimized. In the given input matrix of order n we denote by r_i the entries of row i and by c_j the entries of column j. Our notation means that the numbering of rows is fixed and all entries r_i might be located (moved) at (to) any row j. In order to consider all $n!$ permutations of row entries located at the positions (places) of rows $1,\ldots,n$ these entries will be moved to each possible position. After each movement of entries r_i at the place of row j, this entry contribution to the AP objective function will be computed w.r.t. the given pattern $\mathscr{P}(AP)$. The value of this contribution is simply the sum of all entries appearing in the given pattern $\mathscr{P}(AP)$. We first consider all possible locations of the first row entries r_1 at the positions of rows $1,\ldots,n$. In our example, if the first row entries r_1 are located at the place of row 1 the entry 2 will be located within the cell $(1,1)$ (see Table 3). We will say that the corresponding entry 2 appears in the cell(s) of the given pattern and contributes to its value. After moving the first row entries r_1 at the place of row 2 the entry 8 will be located within the cell $(2,2)$ (see Table 4).

Finally, after moving the first row entries r_1 at the place of row $n = 5$, the entry 3 will be located within the cell $(5,5)$. In other words, by means of locating the entries of r_1 at the places of rows $1,\ldots,n$, the AP pattern will involve each entry of r_1 in all AP feasible solutions. This fact is illustrated in Tables 5 and 6.

If we repeat all movements for all rows entries, then we obtain the so-called *auxiliary matrix* to the original one w.r.t. the AP pattern $\mathscr{P}(AP)$ (see Table 7). As it is easy to see this auxiliary matrix coincides with the original AP matrix and the sum

Table 6 The first row of the auxiliary matrix for the AP

	1	2	3	4	5	
r_1	2	8	4	5	3	1
						2
						3
						4
						5

Table 7 Complete auxiliary matrix for the AP

	1	2	3	4	5
r_1	2	8	4	5	3
r_2	4	9	6	7	1
r_3	3	4	2	0	9
r_4	1	0	9	8	4
r_5	8	3	7	6	5

Table 8 The permuted original matrix by means of an optimal permutation of rows π_1

	1	2	3	4	5	
r_4	1	0	9	8	4	1
r_5	8	3	7	6	5	2
r_1	2	8	4	5	3	3
r_3	3	4	2	0	9	4
r_2	4	9	6	7	1	5

Table 9 All movements of row entries r_1 w.r.t. the pattern $\mathscr{P}(A)$

	1	2	3	4	5	
r_1	2	8	4	5	3	1
r_1	2	8	4	5	3	2
r_1	2	8	4	5	3	3
r_1	2	8	4	5	3	4
r_1	2	8	4	5	3	5

of all entries located within the AP pattern $\mathscr{P}(AP)$ in the original matrix is equal to 26. After solving the AP defined on the pattern $\mathscr{P}(AP)$ we obtain an optimal permutation $\pi_1 = (3,5,4,1,2)$ with its optimal value $a(\pi_1) = 9$ which can be seen explicitly in the permuted matrix (see Table 8) with the sum of all entries at the main diagonal equal to 9.

Our second example deals with the same original AP matrix (see Table 1), but the used pattern is different and defined by the following collection of cells $\mathscr{P}(A) = \{(1,1),(1,2),(2,2),(2,3),(2,4),(3,4),(4,4)\}$ with the sum of all entries in the given pattern equal to 40. The problem in our second example is to find such a permutation of rows that the total sum of all entries within the given pattern $\mathscr{P}(A)$ is minimized. In order to solve this problem we reduce its solution to the usual AP by creating an auxiliary matrix. The auxiliary matrix will be computed if we compute all contributions to the corresponding fragment of our pattern $\mathscr{P}(A)$ for each row entry r_i being located at all possible row positions $j = 1,2,\ldots,5$. All movements of the entries r_1 w.r.t. the pattern A are indicated in Table 9. The corresponding

Table 10 The complete auxiliary matrix to the original one w.r.t. $\mathscr{P}(A)$

	1	2	3	4	5	
r_1	10	17	5	5	0	1
r_2	13	22	7	7	0	2
r_3	7	6	0	0	0	3
r_4	1	17	8	8	0	4
r_5	11	16	6	6	0	5

Table 11 The permuted matrix for pattern $\mathscr{P}(A)$

	1	2	3	4	5	
r_4	1	0	9	8	4	1
r_3	3	4	2	0	9	2
r_1	2	8	4	5	3	3
r_5	8	3	7	6	5	4
r_2	4	9	6	7	1	5

Table 12 Pattern for the LOP

r_1						1
r_2						2
r_3						3
r_4						4
r_5						5

auxiliary matrix for the pattern $\mathscr{P}(A)$ and the same original matrix (after computing all contributions to the different parts of the given pattern when each r_i is located at all places of rows $1,\ldots,5$) are indicated in Table 10.

The entry $a(i,j)$ of the auxiliary matrix shows the contribution to the AP-based model w.r.t. the given pattern $\mathscr{P}(A)$. For example, $a(1,2) = 17$ shows the contribution to the AP objective function w.r.t. the pattern $\mathscr{P}(A)$. This contribution is the sum of all entries appearing within the cells $(2,2),(2,3),\text{and}(2,4)$ after location the entries r_1 at the place of row 2. The complete auxiliary matrix is shown in Table 10 and an optimal permutation of rows $\pi_2 = (3,5,2,1,4)$ with its optimal value $a(\pi_2) = 18$. If we permute all rows of the original matrix by means of the permutation π_2 we will obtain the following permuted matrix (see Table 11) with the sum of all entries at the given pattern $\mathscr{P}(A)$ equal to 18.

The third example is the LOP. The LOP pattern is defined by $\mathscr{P}(LOP) = \{(2,1);(3,1),(3,2);\ldots;(i,1),(i,2),\ldots,(i,i-1);\ldots;(n,1),(n,2),\ldots,(n,n-1)\}$, i.e., all cells (positions) under the main diagonal of the given square matrix of order n (see Table 12). Thus, the LOP is the problem of finding the same permutation for rows and columns such that the objective function (which is the sum of all entries appearing below the main diagonal in the permuted matrix, for example, first by a permutation of rows and second by the same permutation of columns) is minimized.

Our next example is the CFP defined on the Boolean input matrix. In the CFP its optimal pattern $\mathscr{P}(CFP)$ is unknown and should be found together with two optimal permutations: one is a permutation of rows, say r, and another one is a permutation of columns, say c. Thus, the CFP is the problem of finding such a pattern and two permutations, one for rows and another one for columns such that the given objective function is minimized (maximized). Note that in the CFP a feasible pattern consists of an unknown number of block-diagonal-located rectangles with unknown sizes. Let us consider a 5×7 example. For the sake of simplicity we assume that the machine-part Boolean 5×7 matrix and its pattern (the two rectangles with sizes 2×2 and 3×5 are shown by yellow color) are predefined in Table 12. We call the CFP with the given pattern the specified CFP (SCFP). Having a specific pattern we further simplify the SCFP by assuming that the row (machines) order (permutation) is fixed and denote this problem by RSCFP and its pattern by $\mathscr{P}(RSCFP) = \{(1,1),(1,2);(2,1),(2,2);(3,3),(3,4),(3,5),(3,6),(3,7);(4,3),(4,4),(4,5),(4,6),(4,7);(5,3),(5,4),(5,5),(5,6),(5,7)\}$. Informally, we assume that two machine shops are given as follows: $S_1 = \{1,2\}$ and $S_2 = \{3,4,5\}$. Such constraints might be useful in real problems when the machine shops are already built, all the machines are placed inside the shops, and it is too expensive or impossible to move them. Let us also fix the number of parts processed in each shop: two parts in the first shop and five parts in the second one.

Now, in the RSCFP we are given the input matrix, pattern $\mathscr{P}(RSCFP)$, and the fixed order of machines. The RSCFP is the problem of finding a permutation of columns (parts) such that the total sum of all units within (outside) the given pattern $\mathscr{P}(RSCFP)$ is maximized (minimized).

In terms of industrial engineering the Intercell movements are measured by the number of units located outside the cells in the machine-part matrix. Hence, minimizing the intercell movements is equivalent to maximizing the number of units located inside the cells in the machine-part matrix w.r.t. the given pattern $\mathscr{P}(RSCFP)$ (because the total number of units is a constant).

Let us reduce the solution of RSCFP to the minimization version of AP by our pattern-based approach. In order to decide which of two equivalent objective functions we are going to minimize or maximize let us note that the total number of positions within the given pattern is 19 and the number of positions outside the given pattern is $35 - 19 = 16$. It means that in the worst case to compute a contribution to the AP objective function we should sum up the entries at 19 positions inside the given pattern and only 16 entries outside the given pattern. Hence, we have chosen the minimization version of our RSCFP on the complement pattern to the original pattern $\mathscr{P}(RSCFP)$.

Now we are ready to construct an auxiliary square matrix of order 7 since we are going to find out an optimal permutation of seven parts w.r.t. to the given pattern $\mathscr{P}(RSCFP)$. The rows of our auxiliary matrix will be numbered by numbers of columns (parts) of the original matrix and the columns will indicate the contribution $e(i,j)$ to the AP objective function when the column i is located at the place of column j. This contribution $e(i,j)$ is equal to the number of units outside of the given pattern $\mathscr{P}(RSCFP)$, i.e., on the complement to $\mathscr{P}(RSCFP)$ (see Table 14). The complete auxiliary matrix is shown in Table 14 and an optimal permutation of columns $c_1 = (1,3,4,5,6,7,2)$ with its optimal value $RSCFP(c) = 7$. If we permute all columns of the original matrix (see Table 13) by means of the permutation c we

Table 13 The original machine-part 5×7 matrix from Waghodekar and Sahu [49] and its pattern $\mathscr{P}(RSCFP)$ with nine intercell movements

	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 14 The auxiliary matrix for the RSCFP w.r.t. to the given pattern $\mathscr{P}(RSCFP)$

	1	2	3	4	5	6	7
p_1	1	1	1	1	1	1	1
p_2	2	2	1	1	1	1	1
p_3	2	2	1	1	1	1	1
p_4	3	3	1	1	1	1	1
p_5	2	2	2	2	2	2	2
p_6	2	2	1	1	1	1	1
p_7	0	0	1	1	1	1	1

Table 15 The permuted machine-part matrix 5×7 by means zero parts w.r.t the pattern $\mathscr{P}(RSCFP)$ and seven intercell movements

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

obtain the following permuted matrix (see Table 15) with the sum of all entries at the given pattern $\mathscr{P}((RSCFP)$ equal to 7.

As we have mentioned before in the original CFP all three objects, namely, the pattern, the row, and the column permutations, are "decision variables." In the RSCFP we have fixed the pattern and row permutation in the original matrix (see Table 13) and have found an optimal permutation of columns c. Let us fix the found order of parts by means of the permutation c and consider Table 15 as the input matrix for the following CFP. For the given pattern $\mathscr{P}(RSCFP)$ and input matrix (see Table 15) it is necessary to find a permutation of rows (machines) r such that the intercell movements will be further minimized. It means that we are going to construct an auxiliary square matrix of order 5 because we are looking for an optimal permutation of five machines w.r.t. to the given pattern $\mathscr{P}(RSCFP)$. The rows of our auxiliary matrix will be numbered by numbers of machines of the original matrix and their entries will indicate the contribution $d(i,j)$ to the AP objective function when the entries of row i are located at the place of row j. This contribution $d(i,j)$ is equal to the number of units outside of the given pattern $\mathscr{P}(RSCFP)$, i.e., on the complement to $\mathscr{P}(RSCFP)$ (see Table 16). The complete auxiliary matrix is shown in Table 16 and an optimal permutation of rows is $r = (1,4,2,3,5)$ with its optimal value $RSCFP(r) = 5$. If we permute all columns of the original matrix (see Table 15) by means of the permutation r we obtain the following permuted

Table 16 The auxiliary matrix for the RSCFP w.r.t. the permuted matrix (see Table 15) and pattern $\mathscr{P}(RSCFP)$

	1	2	3	4	5
m_1	2	2	2	2	2
m_2	4	4	0	0	0
m_3	4	4	0	0	0
m_4	3	3	1	1	1
m_5	4	4	0	0	0

Table 17 The permuted machine-part matrix 5×7 by means of parts and machines with five intercell movements

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

matrix (see Table 17) with the sum of all entries at the given pattern $\mathscr{P}(RSCFP)$ equal to 5.

In a summary of this section we note that even with the given pattern our AP pattern-based approach to find two independent optimal permutations for rows (machines) and columns (parts) is just a heuristic since our sequential solutions of these two problems are obtained under an assumption that one of the given permutations, say a permutation of rows, is an optimal permutation w.r.t. the unknown another optimal permutation, say a permutation of columns, and vice versa. In the following sections we are going to check whether our heuristic might be competitive with the state-of-the-art heuristics for solving the CFP.

2.2 *The CFP Formulation*

The CFP consists in an optimal grouping of the given machines and parts into cells. The input for this problem is usually given by m machines, p parts, and a rectangular machine-part incidence matrix $A = [a_{ij}]$, where $a_{ij} = 1$ if part j is processed on machine i. The objective is to find an optimal number and configuration of rectangular cells (diagonal blocks in the machine-part matrix) and optimal permutations of rows (machines) and columns (parts) such that after these permutations the number of zeros inside the chosen cells (voids) and the number of ones outside these cells (exceptions) are minimized. Since it is not usually possible to minimize these two values simultaneously there have appeared a number of compound criteria trying to join them into one objective function. Some of them are presented below.

For example, we are given the machine-part matrix shown in Table 13 [49]. Here are two different solutions for this CFP shown in Tables 18 and 19.

The Table 18 solution is better because it has less voids (3 against 4) and exceptions (4 against 5) than the Table 19 solution. But one of its cells is a

Table 18 Solution with singletons

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

Table 19 Solution without singletons

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

singleton – a cell which has less than two machines or products. In some CFP formulations singletons are not allowed, so in this case this solution is not feasible. In this chapter we consider both cases (where singletons are allowed and where they are not allowed) and when there is a solution with singletons found by the suggested heuristic better than without singletons we present both solutions.

2.3 The CFP Objective Functions

There are a number of different objective functions used for the CFP. The following four functions are the most widely used:

1. Grouping efficiency suggested by Chandrasekharan and Rajagopolan (1989):

$$\eta = q\eta_1 + (1-q)\eta_2,$$

where

$$\eta_1 = \frac{n_1 - n_1^{out}}{n_1 - n_1^{out} + n_0^{in}} = \frac{n_1^{in}}{n^{in}},$$

$$\eta_2 = \frac{mp - n_1 - n_0^{in}}{mp - n_1 - n_0^{in} + n_1^{out}} = \frac{n_0^{out}}{n^{out}},$$

η_1 – a ratio showing the intracell loading of machines (or the ratio of the number of ones in cells to the total number of elements in cells).
η_2 – a ratio inverse to the inter-cell movement of parts (or the ratio of the number of zeros out of cells to the total number of elements out of cells).
q – a coefficient ($0 \leq q \leq 1$) reflecting the weights of the machine loading and the intercell movement in the objective function. It is usually taken equal to 0.5,

which means that it is equally important to maximize the machine loading and minimize the intercell movement.
n_1 – a number of ones in the machine-part matrix.
n_0 – a number of zeros in the machine-part matrix.
n^{in} – a number of elements inside the cells.
n^{out} – a number of elements outside the cells.
n_1^{in} – a number of ones inside the cells.
n_1^{out} – a number of ones outside the cells.
n_0^{in} – a number of zeros inside the cells.
n_0^{out} – a number of zeros outside the cells.

2. Grouping efficacy suggested by Kumar and Chandrasekharan [23] to address the drawbacks of the grouping efficiency measure:

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

This measure has proved to better reflect the quality of a CFP solution.
3. Group capability index (GCI) suggested by Hsu [18]:

$$GCI = 1 - \frac{n_1^{out}}{n_1} = \frac{n_1 - n_1^{out}}{n_1}.$$

4. Number of exceptions (ones outside cells) and voids (zeros inside cells):

$$E + V = n_1^{out} + n_0^{in}.$$

In this chapter we use the grouping efficacy measure in all the computational experiments because of its capability to distinguish good and bad solutions and other useful properties (see papers of Kumar and Chandrasekharan [23] and Goncalves and Resende [17] for more information). To show the difference between the four described objective functions we calculate these values for the two solutions presented above in Tables 18 and 19.

$$\eta = 0.5\frac{16}{19} + 0.5\frac{12}{16} \approx 79.60\% \qquad \eta = 0.5\frac{15}{19} + 0.5\frac{11}{16} \approx 73.85\%$$
$$\tau = \frac{20-4}{20+3} \approx 69.57\% \qquad \tau = \frac{20-5}{20+4} \approx 62.50\%$$
$$GCI = \frac{20-4}{20} \approx 80.00\% \qquad GCI = \frac{20-5}{20} \approx 75.00\%$$
$$E + V = 4 + 3 = 7 \qquad E + V = 5 + 4 = 9$$

Claim. If the pattern (the number and configuration of the cells) is fixed then objective functions η, τ, GCI, $E+V$ become equivalent, in other words these functions reach their optimal values on the same solution.

Proof. For the fixed pattern the following values are constant: n_1, n_0, n^{in}, n^{out}. So if we maximize the number of ones inside the pattern n_1^{in} then $n_0^{in} = n^{in} - n_1^{in}$ is minimized, $n_0^{out} = n_0 - n_0^{in}$ is maximized, and $n_1^{out} = n_1 - n_1^{in}$ is minimized. This means that the grouping efficiency $\eta = q\frac{n_1^{in}}{n^{in}} + (1-q)\frac{n_0^{out}}{n^{out}}$ is maximized, the grouping efficacy $\tau = \frac{n_1^{in}}{n_1+n_0^{in}}$ is maximized, the grouping capability index $GCI = 1 - \frac{n_1^{out}}{n_1}$ is maximized, and the number of exceptions plus voids $E + V = n_1^{out} + n_0^{in}$ is minimized. □

That is why when we apply the pattern-based approach to find optimal permutations for the fixed pattern we maximize the sum of elements inside the pattern which is equal to n_1^{in}.

3 Heuristic

In this section we describe and demonstrate the suggested pattern-based approach on the same 5x7 example of Waghodekar and Sahu [49] which we already used in the previous section (Table 20). Basic steps of the suggested algorithm are the following:

1. Pattern-based heuristic
   At the first stage our goal is to find the optimal number of cells k^* for the current problem instance for which we will then generate different input patterns and obtain solutions starting from these patterns on the next stage. The algorithm is the following:

   (a) Choose a number of cells k.
   We try numbers of cells in the range from 2 to $\min(m,p)/2$ where m is the number of rows (machines) in the machine-part matrix and p is the number of columns (parts).

   (b) Build an initial pattern. For the chosen number of cells k we build an initial pattern in the following way. The rows and columns are divided into equal blocks of $[m/k]$ rows and $[p/k]$ columns (here by $[x]$ we denote an integer part of x). The diagonal blocks are the cells for the CFP. If m or p is not divisible by k then all the remaining rows and columns are added to the last cell. An initial pattern with two cells for our example is shown in Table 21.

Table 20 Machine-part 5x7 matrix from Waghodekar and Sahu [49]

	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 21 Initial pattern with two cells

	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 22 Auxiliary matrix for rows

	1	2	3	4	5
1	1	1	3	3	3
2	2	2	2	2	2
3	1	1	3	3	3
4	3	3	1	1	1
5	1	1	3	3	3

Table 23 Modified auxiliary matrix for rows

	1	2	3	4	5
1	2	2	0	0	0
2	1	1	1	1	1
3	2	2	0	0	0
4	0	0	2	2	2
5	2	2	0	0	0

Table 24 Optimal AP solution (optimal positions of rows)

	1	2	3	4	5
1	2	2	0	0	0
2	1	1	1	1	1
3	2	2	0	0	0
4	0	0	2	2	2
5	2	2	0	0	0

Note that this pattern is different from the pattern considered in the previous section and so the result will be different. Moreover, the returned by our heuristic solutions depend on the order of permuted either rows or columns of the input matrix. In the previous section we first have found a permutation of columns c_1 and after permuting the original input matrix by means of c_1 found a permutation of rows $r_1 = r_1(c_1)$ which depends on the found permutation of columns c_1.

(c) Form an auxiliary matrix for rows (Table 22).
(d) Modify the auxiliary matrix ($a_{ij} = \max_{k,l} a_{kl} - a_{ij}$) to obtain a minimization problem (Table 23).
(e) Solve the AP for this matrix and obtain an AP optimal rows permutation (Table 24).
(f) Permute rows of the original machine-part matrix according to the optimal AP permutation from step e (Table 25).

Table 25 Machine-part matrix after the optimal row permutation

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

Table 26 Auxiliary matrix for columns

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	2	2	2	1	1	1	1
3	2	2	2	1	1	1	1
4	2	2	2	2	2	2	2
5	1	1	1	3	3	3	3
6	0	0	0	3	3	3	3
7	0	0	0	1	1	1	1

Table 27 Modified auxiliary matrix for columns

	1	2	3	4	5	6	7
1	2	2	2	2	2	2	2
2	1	1	1	2	2	2	2
3	1	1	1	2	2	2	2
4	1	1	1	1	1	1	1
5	2	2	2	0	0	0	0
6	3	3	3	0	0	0	0
7	3	3	3	2	2	2	2

Table 28 Optimal AP solution (optimal positions of columns)

	1	2	3	4	5	6	7
1	2	2	2	2	2	2	2
2	1	1	1	2	2	2	2
3	1	1	1	2	2	2	2
4	1	1	1	1	1	1	1
5	2	2	2	0	0	0	0
6	3	3	3	0	0	0	0
7	3	3	3	2	2	2	2

(g) Form an auxiliary matrix for columns based on the permuted machine-part matrix from the previous step (Table 26).

(h) Modify the auxiliary matrix ($a_{ij} = \max_{k,l} a_{kl} - a_{ij}$) to get a minimization problem (Table 27).

(i) Solve the AP for this matrix and obtain an AP optimal column permutation (Table 28).

Table 29 Moving part 4 from cell 2 to cell 1

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

(j) Permute columns according to the AP optimal permutation from the previous step. In this case we have got an identical (trivial) permutation, so the columns should stay on their places and the machine-part matrix remains the same as shown in Table 25.

2. Pattern-modification improvement heuristic

(k) Apply the pattern-modification improvement heuristic to improve the solution found so far. The main idea of the improvement heuristic is that the grouping efficacy can usually be increased by simple modifications (moving either a row or a column from one cell to another) of the current pattern (cell configuration). To compute the grouping efficacy for the obtained solution (Table 25) we need the total number of ones n_1, the number of zeros inside the cells n_0^{in}, and the number of ones outside the cells n_1^{out}: $n_1 = 20,\ n_0^{in} = 4,$ and $\mathrm{n}_1^{\mathrm{out}} = 6$. The grouping efficacy is then calculated by the following formula:

$$\tau = \frac{n_1 - n_1^{out}}{n_1 + n_0^{in}} = \frac{20-6}{20+4} \approx 58.33\,\%.$$

Looking at this solution (Table 25) we can conclude that it is possible to move part 4 from the second cell to the first one. And this way the number of zeros inside cells decreases by 1 and the number of ones outside cells remains the same. So it is profitable to attach column 4 to the first cell as it is shown in Table 29.

For the modified pattern we have $n_0^{in} = 3,\ n_1^{out} = 6$, and the grouping efficacy:

$$\tau = \frac{20-6}{20+3} \approx 60.87\,\%.$$

As a result the efficacy is increased by 2.5 percent. Computational results show that using such pattern modifications could considerably improve the solution. The idea is to compute an increment in efficacy for each column and row when it is moved to all other cells and then perform the modification corresponding to the maximal increment (Tables 30 and 31).

Table 30 Efficacy after moving a row to another cell

	Efficacy	
Row	Cell 1 (%)	Cell 2 (%)
1	58.33	56.00
2	58.33	44.44
3	48.00	58.33
4	48.00	58.33
5	48.00	58.33

Table 31 Efficacy after moving a column to another cell

	Efficacy	
Column	Cell 1 (%)	Cell 2 (%)
1	58.33	56.00
2	58.33	50.00
3	58.33	50.00
4	60.87	58.33
5	48.00	58.33
6	42.31	58.33
7	54.17	58.33

We make a modification which gives the maximal increase in efficacy within all the results – for rows and for columns. Looking at Tables 30 and 31 we can conclude that only moving part 4 to cell 1 could increase the grouping efficacy of the solution. Such modifications are repeated until there is still a column or a row for which we get an increment in efficacy.

Note that here we have obtained the solution with efficacy 60.87 % which is different from the solution shown in the previous chapter with efficacy 62.50 %. This is because here we use another pattern and also we find an optimal permutation first for rows and then for columns.The order of these steps influences the solution found by pattern-based heuristic. That is why we then repeat the same procedure (steps c–k), but first we find an optimal permutation for columns and then for rows. Since the number of columns (parts) is usually greater than the number of rows (machines) then the number of possible column permutations is much greater than the number of possible row permutations. This means that we have a greater flexibility for column permutation and usually when starting from columns we obtain a better solution. Anyway, we try both row-column and column-row orders of permutations and then choose the best solution.

(l) Repeat steps a–k for different numbers of cells from 2 to $\min(m,p)/2$.

3. Determine the number of cells k^* for which the best solution is obtained.

(m) After we have found the grouping efficacy for first solutions with different numbers of cells we compare them and choose the optimal cell number – k^*, for which the greatest grouping efficacy is obtained.

4. Generate additional patterns with $k^* - c$, $k^* - c+1, \ldots, k^* + c$ cells.

(n) We enumerate patterns with cell number taken from a small c-neighborhood of k^*. Solutions without singletons usually do not require a big number of cells because height and width of every cell of a solution cannot be less than two. So there are no such a variety of possible patterns as it is for solutions with singletons. That is why we take $c = 2$ for solutions without singletons and $c = 5$ for solutions with singletons. The next step is repeated for every number of cells from $k^* - c$ to $k^* + c$ (2c+1 times).

(o) For the fixed number of cells we generate different patterns enumerating different values for width and height of every cell with a step of two units (a step of one unit is used only for matrices smaller than 15x20). It means that for solutions without singletons we generate only cells with width and height equal to 3, 5, …, except the last cell which can have an even width or height. For solutions with singletons we use 2, 4, … values. The step of two units is explained by the fact that our improvement heuristic (which is then applied to every pattern) makes elementary modifications of patterns and moves one row or one column from one cell to another if it increases the grouping efficacy.

Since our pattern-based heuristic permutes rows and columns of the machine-part matrix then the order of the cells in the pattern does not matter. So we can generate cells so that the first cell has the smallest dimensions and the last one has the greatest.

For example, if we have a 30x50 machine-part matrix and we want to generate patterns with five cells for non-singleton solutions, then we proceed as follows. First, we generate all possible combinations of cell heights with the difference between each pair of sequential neighboring cells equal to 2 (we call this difference by size increment). There are 13 possible combinations for this example. Second, we generate all possible combinations of cell widths with size increment 2. There are 119 possible combinations for this example. Third, we combine every height combination with every width combination to form a pattern. So we get $13 * 119 = 1,547$ different patterns for this example.

- Combinations of heights:

1. $3+3+3+3+18 = 30$
2. $3+3+3+5+16 = 30$
3. $3+3+3+7+14 = 30$
4. $3+3+3+9+12 = 30$
5. $3+3+5+5+14 = 30$
6. $3+3+5+7+12 = 30$
7. $3+3+5+9+10 = 30$
8. $3+3+7+7+10 = 30$
9. $3+5+5+5+12 = 30$
10. $3+5+5+7+10 = 30$

11. $3+5+7+7+8=30$
12. $5+5+5+5+10=30$
13. $5+5+5+7+8=30$

- Combinations of widths:

  1. $3+3+3+3+38=50$
  2. $3+3+3+5+36=50$

     ...

  119. $9+9+9+11+12=50$

- Patterns:

  1. $3\times 3, 3\times 3, 3\times 3, 3\times 3, 18\times 38$

     ...

  119. $3\times 9, 3\times 9, 3\times 9, 3\times 11, 18\times 12$
  120. $3\times 3, 3\times 3, 3\times 3, 5\times 3, 16\times 38$

     ...

  1547. $5\times 9, 5\times 9, 5\times 9, 7\times 11, 8\times 12$

We then apply our heuristics using all these patterns as initial on the next steps of the algorithm.

5. Run the pattern-based and improvement heuristics for all the patterns and choose the best found solution.

   (p) Steps a–k are repeated for all patterns generated on steps n–o. The best found solution is then taken as the heuristic solution to the CFP.

4 Computational Results

For our computational experiments with the pattern-based heuristic (PBH) we selected the most popular 35 GT instances from the literature (see, e.g., Goncalves and Resende [17]). We compare our solutions to the 35 GT instances in terms of the grouping efficacy with best solutions reported up to date of this paper submission. The currently best heuristic for the CFP is the evolutionary algorithm (EA) from Goncalves and Resende [17]. Since allowing singletons (cells with only one machine or only one part) in a CFP solution is arguable we present both solutions (with singletons and without) in all cases where our heuristic has been able to find a better solution with singletons. Also in our solutions we forbid parts which are not included to any cell, though in many cases it is more efficient (in terms of the grouping efficacy) to leave some parts not assigned to any cell.

In Table 32 we compare our PBH with EA heuristic. We do not include the results of other six approaches (ZODIAC by Chandrasekharan and Rajagopalan [12], GRAFICS by Srinivasan and Narendran [46], MST—clustering algorithm by

Table 32 Comparison of our results with EA algorithm [17]

#	Source	Size	Our approach				EA	
			Singletons		No singletons		No singletons	
			Cells	Efficacy	Cells	Efficacy	Cells	Efficacy
1	King and Nakornchai [21]	5 × 7	3	**75.00**	2	73.68	2	73.68
2	Waghodekar and Sahu [49]	5 × 7	2	**69.57**	2	62.50	2	62.50
3	Seifoddini [42]	5 × 18	2	79.59[d]	2	79.59	2	79.59
4	Kusiak (1992) [29]	6 × 8	2	76.92[d]	2	76.92	2	76.92
5	Kusiak and Chow [28]	7 × 11	5	**60.87**	3	53.13	3	53.13
6	Boctor [4]	7 × 11	4	**70.83**	3	70.37	3	70.37
7	Seifoddini and Wolfe [43]	8 × 12	4	**69.44**	3	68.29	3	68.29[a]
8	Chandrasekharan and Rajagopalan [11]	8 × 20	3	85.25[d]	3	85.25	3	85.25
9	Chandrasekharan and Rajagopalan [10]	8 × 20	2	58.72[d]	2	58.72	2	58.72
10	Mosier and Taube [36]	10 × 10	5	**75.00**	3	70.59	3	70.59
11	Chan and Milner [9]	10 × 15	3	92.00[d]	3	92.00	3	92.00
12	Askin and Subramanian [2]	14 × 23	6	**75.00**	5	69.86	5	69.86
13	Stanfel [48]	14 × 24	7	**71.83**	5	69.33	5	69.33
14	McCormick et al. [32]	16 × 24	7	**53.76**	6	51.96	6	51.96[b]
15	Srinivasan et al. [47]	16 × 30	6	**68.99**	4	67.83	4	67.83
16	King [20]	16 × 43	8	**57.53**	6	**55.83**	5	54.86
17	Carrie [8]	18 × 24	9	**57.73**	6	54.46	6	54.46
18	Mosier and Taube [37]	20 × 20	5	**43.45**	5	**42.96**	5	42.94
19	Kumar et al. [24]	20 × 23	7	**50.81**	5	49.65	5	49.65
20	Carrie [8]	20 × 35	5	**78.40**	5	**76.54**	4	76.14[c]

(continued)

Table 32 (continued)

#	Source	Size	Our approach				EA	
			Singletons		No singletons		No singletons	
			Cells	Efficacy	Cells	Efficacy	Cells	Efficacy
21	Boe and Cheng [5]	20×35	5	**58.38**	5	**58.15**	5	58.07
22	Chandrasekharan and Rajagopalan [13]	24×40	7	100.00[d]	7	100.00	7	100.00
23	Chandrasekharan and Rajagopalan [13]	24×40	7	85.11[d]	7	85.11	7	85.11
24	Chandrasekharan and Rajagopalan [13]	24×40	7	73.51[d]	7	73.51	7	73.51
25	Chandrasekharan and Rajagopalan [13]	24×40	11	**53.29**	10	**51.97**	9	51.88
26	Chandrasekharan and Rajagopalan [13]	24×40	12	**48.95**	10	**47.37**	9	46.69
27	Chandrasekharan and Rajagopalan [13]	24×40	12	**46.26**	10	**44.87**	9	44.75
28	McCormick et al. [32]	27×27	5	**54.82**	4	54.27	4	54.27
29	Carrie [8]	28×46	11	**47.23**	11	**45.92**	9	44.37
30	Kumar and Vannelli [25]	30×41	15	**62.77**	12	**58.94**	11	58.11
31	Stanfel [48]	30×50	13	**59.77**	12	**59.66**	12	59.21
32	Stanfel [48]	30×50	14	**50.83**	11	**50.51**	11	50.48
33	King and Nakornchai [21]	30×90	16	**48.01**	11	**45.74**	9	42.12
34	McCormick et al. [32]	37×53	3	**60.50**	2	**59.29**	2	56.42
35	Chandrasekharan and Rajagopalan [12]	40×100	10	84.03[d]	10	84.03	10	84.03

[a] In Goncalves and Resende [17] the result of 68.30 is presented though it is actually 68.29 (calculated using the solution presented in the appendix of that paper)
[b] In Goncalves and Resende [17] the result of 52.58 is presented though it is actually 51.96 (calculated using the solution presented in the appendix of that paper)
[c] In Goncalves and Resende [17] the result of 76.22 is presented though it is actually 76.14 (calculated using the solution presented in the appendix of that paper)
[d] This solution actually does not have any singletons but it was the best solution found by the algorithm with allowed singletons

Table 33 Corrections of the grouping efficacy values published in Goncalves and Resende [17]

#	Source	Problem size	Reported value	Corrected value
7	Seifoddini and Wolfe [43]	8 × 12	68.30	68.29
14	McCormick et al. [32]	16 × 24	52.58	51.96
20	Carrie [8]	20 × 35	76.22	76.14

Srinivasan [45], GATSP—genetic algorithm by Cheng et al. [14], GA—genetic algorithm by Onwubolu and Mutingi [40], GP—genetic programming by Dimopoulos and Mort [15]) also considered in the work of Goncalves and Resende [17] because EA has the best results among all these approaches on all GT 35 instances. Note that some of grouping efficacy values published in Goncalves and Resende [17] do not correspond to their found solutions shown in the appendix of that paper. So we present the corrected values for the EA algorithm in Table 32. In Table 33 we show the corrections which we have made.

As it can be seen from Table 32 all our solutions are better or equal to the EA solutions (better results are shown with bold font). More specifically we have improved the grouping efficacy for 13 instances and found solutions with the same value of grouping efficacy for the remaining 22 instances. For these 22 instances we have the same efficacy, but the solutions are different (see appendix). The maximum improvement is 7 % for 37x53 instance of McCormick et al. [32]; the average improvement among the 13 improved instances is 1.8 %. The solutions with singletons are better than without them by 2.6 % in average. A short summary of comparison is shown in Table 34. For 26 of the 35 instances the algorithm has found a solution with singletons which is better than without. Only for three of these instances the solution has the same number of cells as the solution of the EA algorithm without singletons. All the 22 solutions without singletons which have the same grouping efficacy with the solutions of the EA algorithm also have the same number of cells, though the configuration of the cells and the distributions of ones and zeros are different. The 13 solutions without singletons which are better than EA algorithm solutions usually have more cells (eight solutions against five solutions with the same number of cells).

5 Summary and Future Research Directions

In this chapter we present a new pattern based approach within the classic linear assignment model.The main idea of this approach might be illustrated by means of different classes of COPs which include the maximum weight independent set and its unweighted version, linear ordering, and cell formation problems, just to mention

Table 34 Summary of the comparison with Goncalves and Resende [17]

Number of problems solved													
Better						Equal						Total	
No singletons # of cells			Singletons # of cells			No singletons # of cells			Singletons # of cells			No singletons	Singletons
More	Equal	Total	More	Equal	Total	More	Equal	Total	More	Equal	Total	35	26
8	5	**13**	23	3	**26**	0	22	22	0	0	0		

a few. We have successfully applied this approach to design a new heuristic which outperforms all well-known heuristics for the CFP with the grouping efficacy as its objective function. The main distinctions of our PBH are as follows:

1. The PBH is based on a rigorous (even if it is informal) formulation of the CFP as the problem of finding three objects, namely, (a) an optimal pattern, (b) an optimal part permutation, and (c) an optimal machine permutation.
2. Our rigorous formulation might be solved efficiently for any fixed pattern and permutation (either parts or machines) by means of the Jonker–Volgenant's Hungarian algorithm [20] efficient implementation.
3. Based on our formulation of the CFP we have designed an efficient PBH which outperforms all currently known heuristics for the CFP with the grouping efficacy criterion of optimality.
4. We believe that the success of our PBH is due to a wide range of patterns sequentially enumerated under control of the optimality criterion.
5. Since to solve a CFP instance, say 40x100, with a specific pattern by our PBH requires just several milliseconds, we are able to involve much more adjusted patterns than we have done in this study and hence to generate a wider range of high quality CFP solutions.

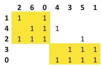
Our main research direction will be concentrated on the exact mathematical programming formulation of the CFP with the purpose to find out the thresholds for the number of machines and parts which might be treated to optimality within the mathematical programming including fractional programming approach. Another direction of our research will be in finding polynomially solvable special cases of the CFP based either on structural properties of the Boolean input machine-part matrix or the CFP criteria of optimality. Finally we are looking for applications of the pattern based approach to other classes of COPs.

Acknowledgments The authors would like to thank professor Mauricio Resende for the 35 GT instances provided on his web page (http://www2.research.att.com/~mgcr/data/cell-formation/). We are also grateful to professors R. Jonker and A. Volgenant for making available for us their very efficient program implementation of the Hungarian algorithm.

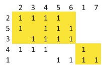
The authors are supported by The LATNA Laboratory, National Research University Higher School of Economics (NRU HSE), and Russian Federation government grant, ag. 11.G34.31.0057. Boris Goldengorin and Mikhail Batsyn are partially supported by NRU HSE Scientific Fund grant "Teachers–Students" #11-04-0008 "Calculus for tolerances in combinatorial optimization: theory and algorithms."

Appendix: Pattern-Based Solutions Reported to Table 32

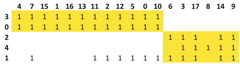
Without Singletons



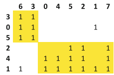
1. King and Nakornchai (1982), size 5x7, 2 cells, efficacy 73.68



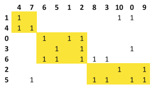
2. Waghodekar and Sahu (1984), size 5x7, 2 cells, efficacy 62.50



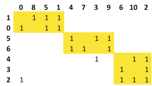
3. Seifoddini (1989), size 5x7, 2 cells, efficacy 79.59



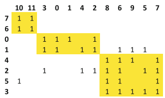
4. Kusiak (1992), size 6x8, 2 cells, efficacy 76.92



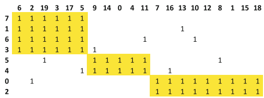
5. Kusiak and Chow (1987), size 5x7, 3 cells, efficacy 53.13



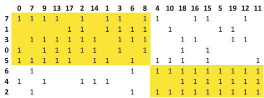
6. Boctor (1991), size 7x11, 3 cells, efficacy 70.37



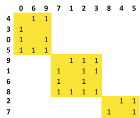
7. Seifoddini and Wolfe (1986), size 8x12, 3 cells, efficacy 68.29



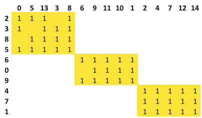
8. Chandrasekharan and Rajagopalan (1986b), size 8x20, 3 cells, efficacy 85.25



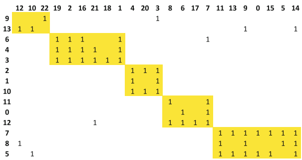
9. Chandrasekharan and Rajagopalan (1986a), size 8x20, 2 cells, efficacy 58.72



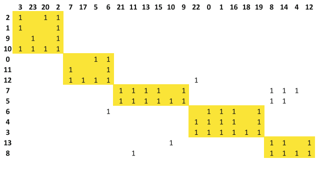
10. Mosier and Taube (1985a), size 10x10, 3 cells, efficacy 70.59



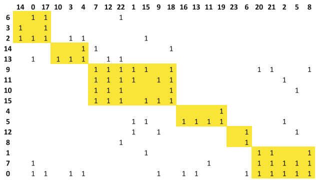
11. Chan and Milner (1982), size 10x15, size 10x15, 3 cells, efficacy 92.00



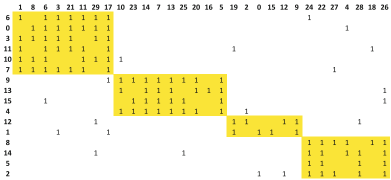
12. Askin and Subramanian (1987), size 14x23, 5 cells, efficacy 69.86



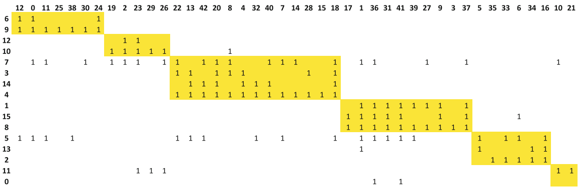
13. Stanfel (1985), size 14x24, 5 cells, efficacy 69.33



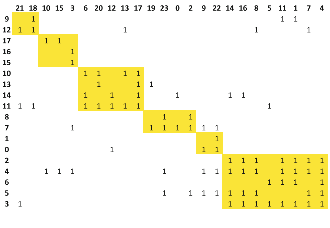
14. McCormick et al. (1972), size 16x24, 6 cells, efficacy 51.96



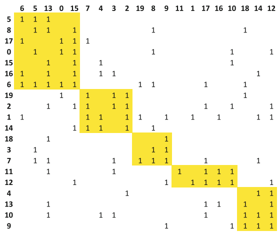
15. Srinivasan et al. (1990), size 16x30, 4 cells, efficacy 67.83



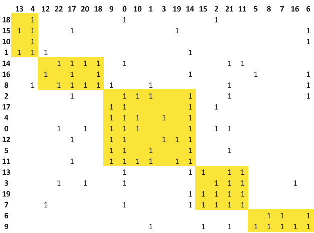
16. King (1980), size 16x43, 6 cells, efficacy 55.83



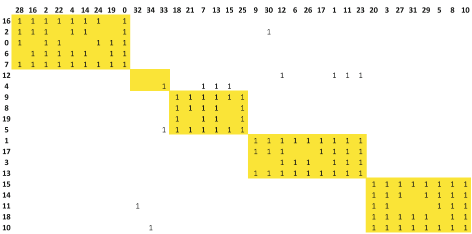
17. Carrie (1973), size 18x24, 6 cells, efficacy 54.46



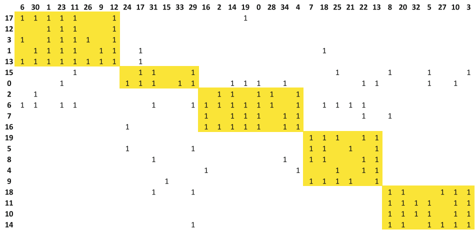
18. Mosier and Taube (1985b), size 20x20, 5 cells, efficacy 42.96



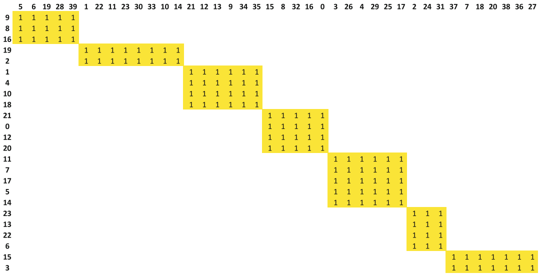
19. Kumar et al. (1986), size 20x23, 5 cells, efficacy 49.65



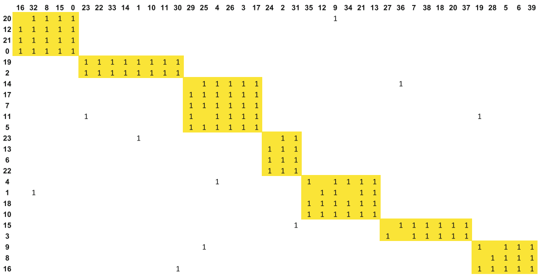
20. Carrie (1973), size 20x35, 5 cell, efficacy 76.54



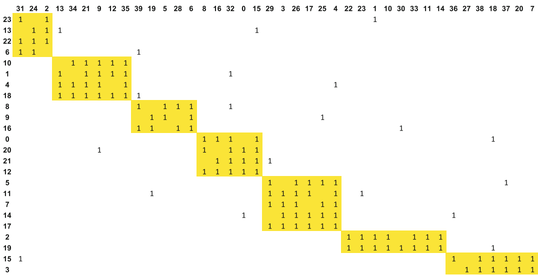
21. Boe and Cheng (1991), size 20x35, 5 cells, efficacy 58.15



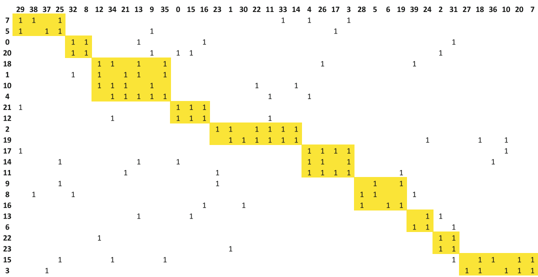
22. Chandrasekharan and Rajagopalan (1989), size 24x40, 7 cells, efficacy 100.00



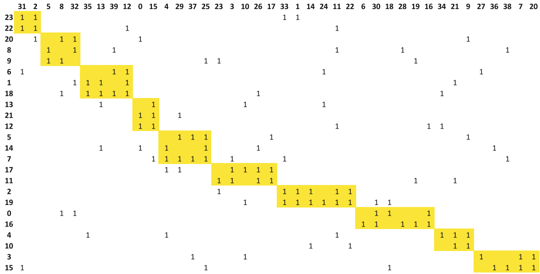
23. Chandrasekharan and Rajagopalan (1989), size 24x40, 7 cells, efficacy 85.11



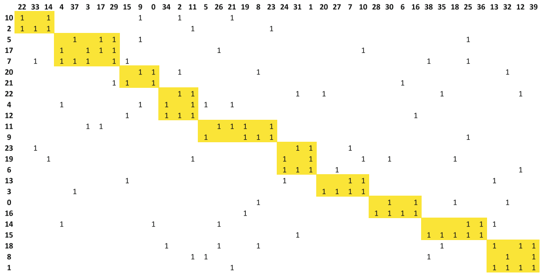
24. Chandrasekharan and Rajagopalan (1989), size 24x40, 7 cells, efficacy 73.51



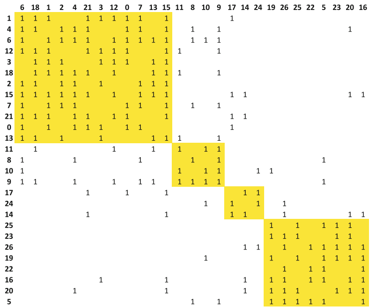
25. Chandrasekharan and Rajagopalan (1989), size 24x40, 10 cells, efficacy 51.97



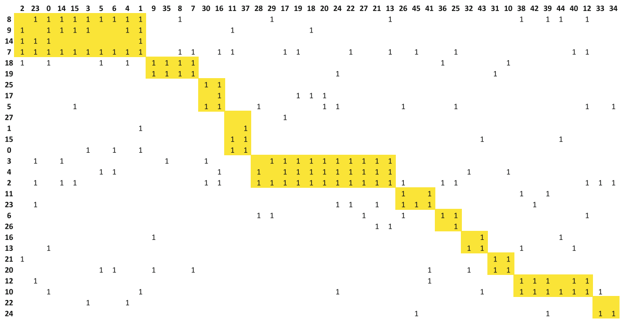
26. Chandrasekharan and Rajagopalan (1989), size 24x40, 10 cells, efficacy 47.37



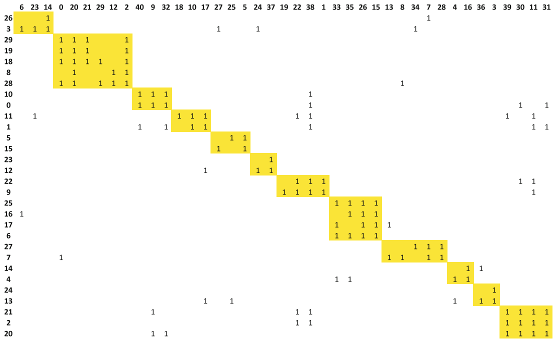
27. Chandrasekharan and Rajagopalan (1989), size 24x40, 10 cells, efficacy 44.87



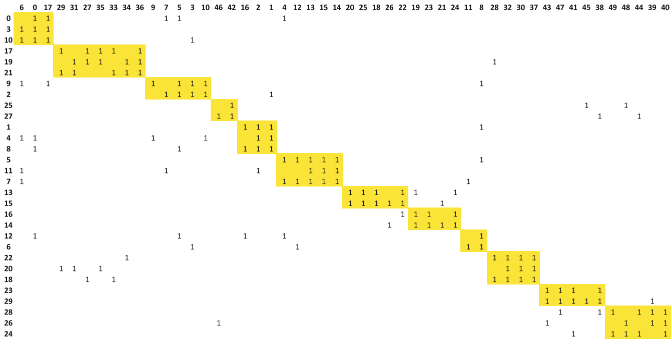
28. McCormick et al. (1972), size 27x27, 4 cells, efficacy 54.27



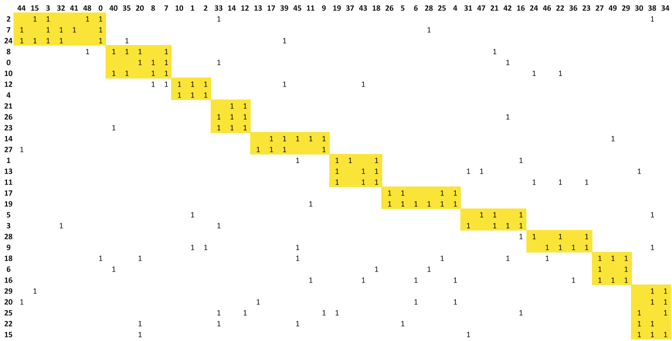
29. Carrie (1973), size 28x46, 11 cells, efficacy 45.92



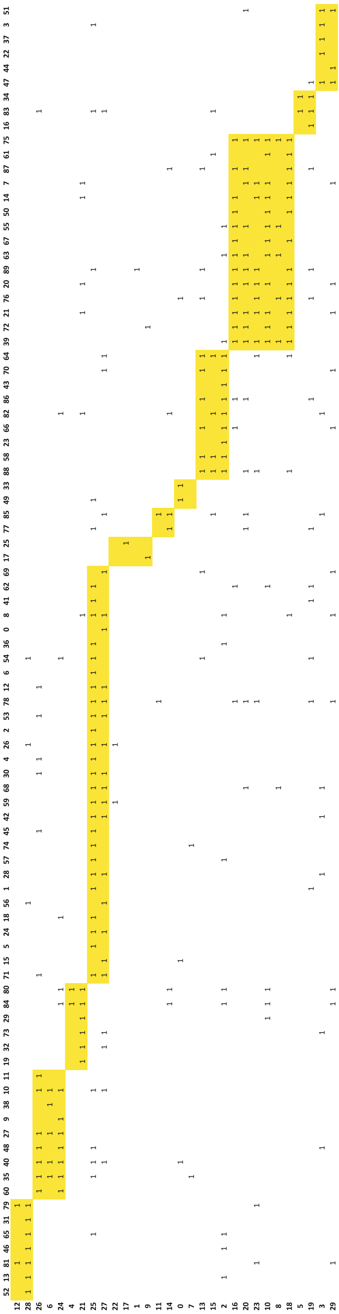
30. Kumar and Vannelli (1987), size 30x41, 12 cells, efficacy 58.94



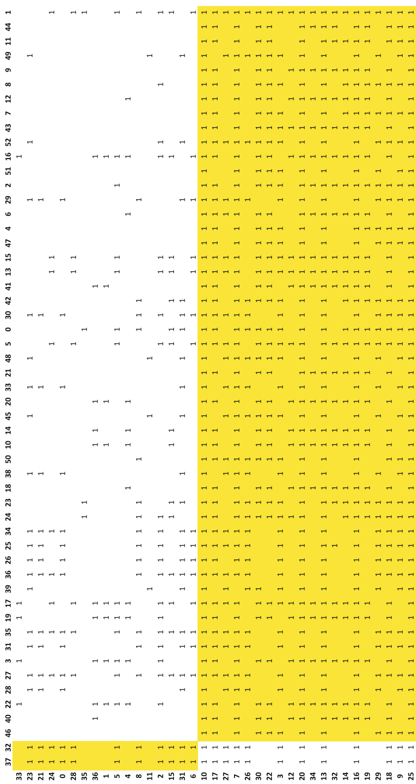
31. Stanfel (1985), size 30x50, 12 cells, efficacy 59.66



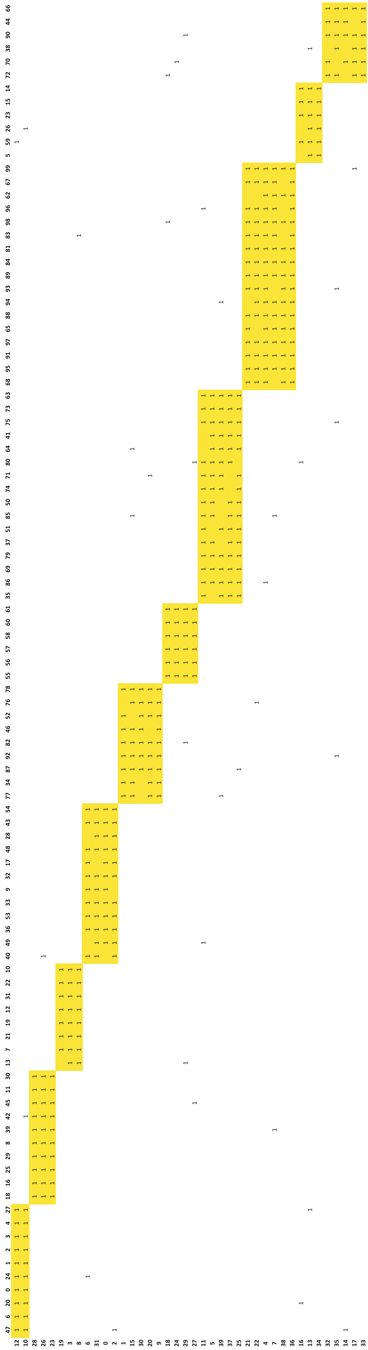
32. Stanfel (1985), size 30x50, 11 cells, efficacy 50.51



33. King and Nakornchai (1982), size 30x90, 11 cells, efficacy 45.74

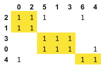


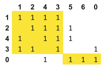
34. McCormick et al. (1972), size 37x53, 2 cells, efficacy 59.29

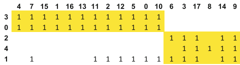


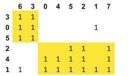
35. Chandrasekharan and Rajagopalan (1987), size 40x100, efficacy 84.03

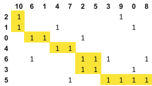
With Singletons

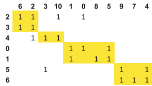

1. King and Nakornchai (1982), size 5x7, 3 cells, efficacy 75.00

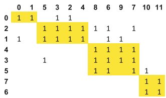

2. Waghodekar and Sahu (1984), size 5x7, 2 cells, efficacy 69.57

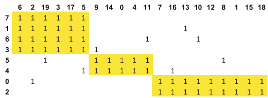

3. Seifoddini (1989), size 5x7, 2 cells, efficacy 79.59

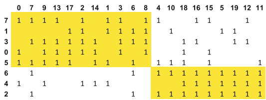

4. Kusiak (1992), size 6x8, 2 cells, efficacy 76.92

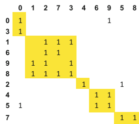

5. Kusiak and Chow (1987), size 5x7, 5 cells, efficacy 60.87

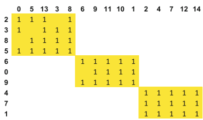

6. Boctor (1991), size 7x11, 4 cells, efficacy 70.83

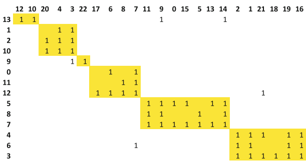

7. Seifoddini and Wolfe (1986), size 8x12, 4 cells, efficacy 69.44

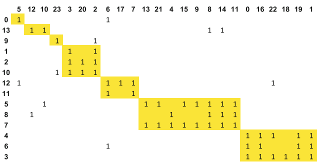

8. Chandrasekharan and Rajagopalan (1986b), size 8x20, 3 cells, efficacy 85.25

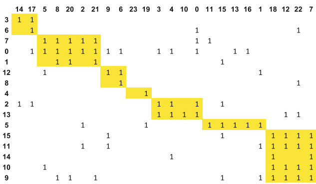

9. Chandrasekharan and Rajagopalan (1986a), size 8x20, 2 cells, efficacy 58.72

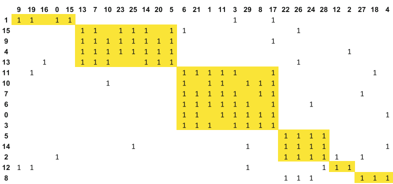

10. Mosier and Taube (1985a), size 10x10, 5 cells, efficacy 75

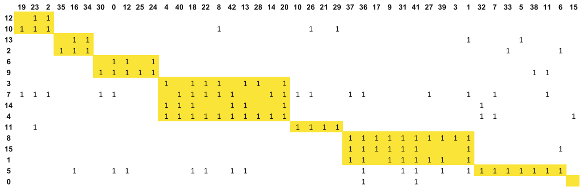

11. Chan and Milner (1982), size 10x15, size 10x15, 3 cells, efficacy 92.00


12. Askin and Subramanian (1987), size 14x23, 6 cells, efficacy 75.00

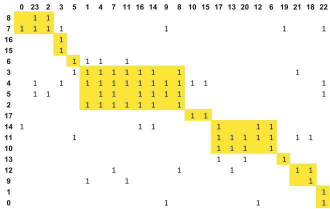

13. Stanfel (1985), size 14x24, 7 cells, efficacy 71.83


14. McCormick et al. (1972), size 16x24, 7 cells, efficacy 53.76

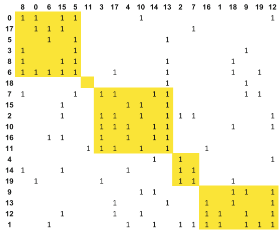

15. Srinivasan et al. (1990), size 16x30, 6 cells, efficacy 68.99



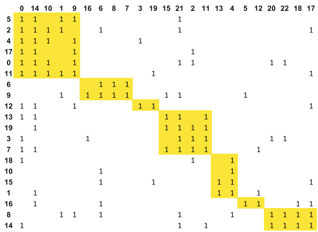
16. King (1980), size 16x43, 8 cells, efficacy 57.53



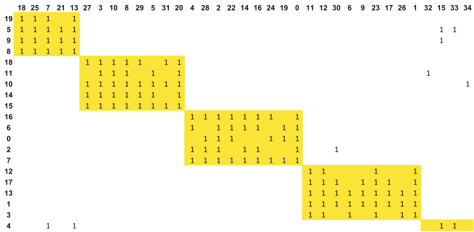
17. Carrie (1973), size 18x24, 9 cells, efficacy 57.73



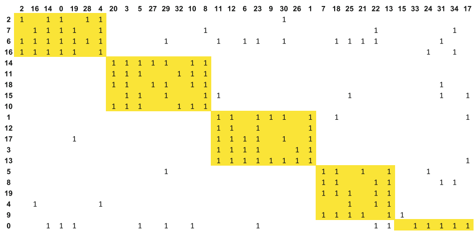
18. Mosier and Taube (1985b), size 20x20, 5 cells, efficacy 43.45



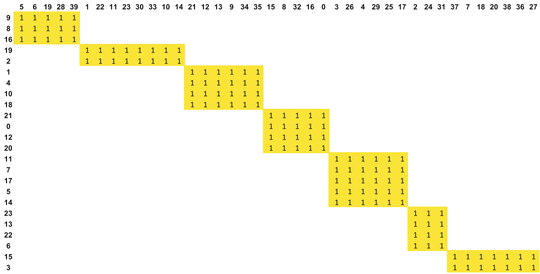
19. Kumar et al. (1986), size 20x23, 7 cells, efficacy 50.81



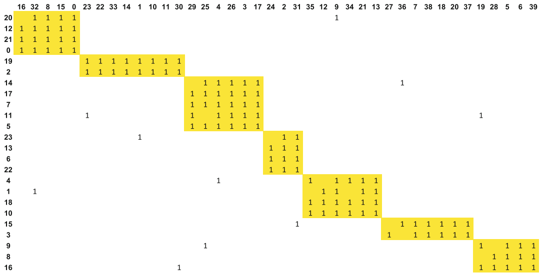
20. Carrie (1973), size 20x35, 5 cell, efficacy 78.40



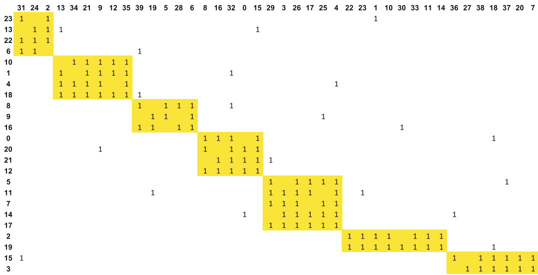
21. Boe and Cheng (1991), size 20x35, 5 cells, efficacy 58.38



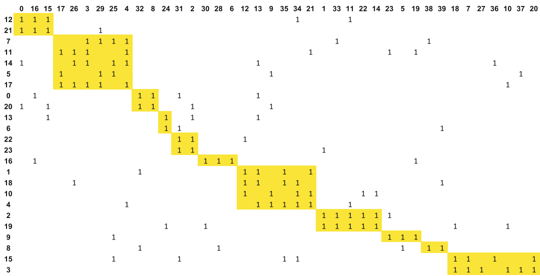
22. Chandrasekharan and Rajagopalan (1989), size 24x40, 7 cells, efficacy 100.00



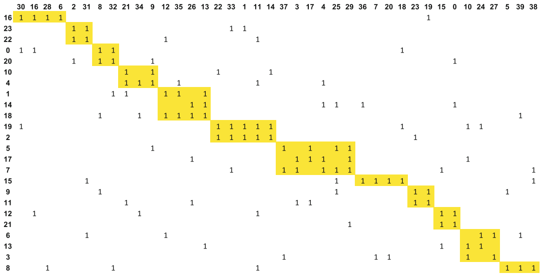
23. Chandrasekharan and Rajagopalan (1989), size 24x40, 7 cells, efficacy 85.11



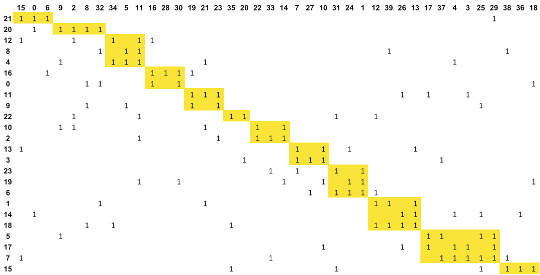
24. Chandrasekharan and Rajagopalan (1989), size 24x40, 7 cells, efficacy 73.51



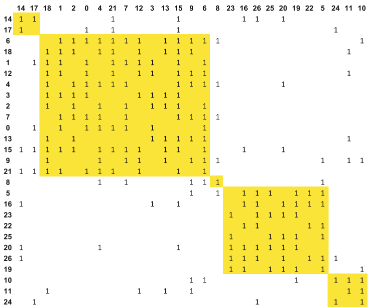
25. Chandrasekharan and Rajagopalan (1989), size 24x40, 11 cells, efficacy 53.29



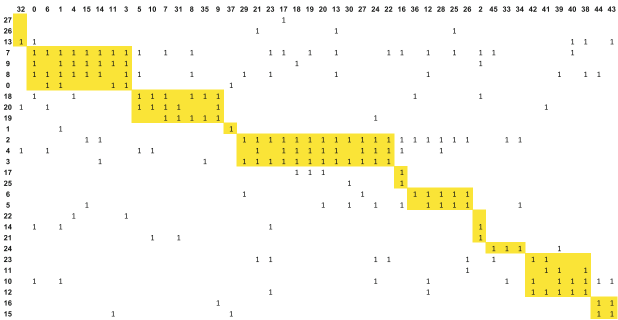
26. Chandrasekharan and Rajagopalan (1989), size 24x40, 12 cells, efficacy 48.95



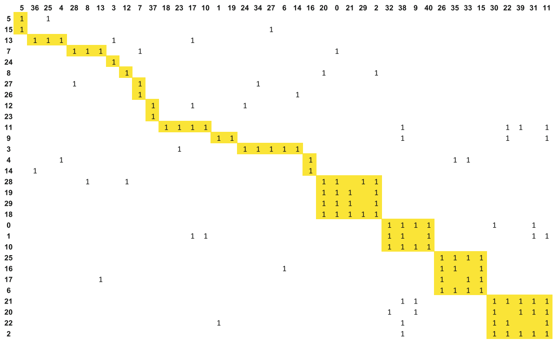
27. Chandrasekharan and Rajagopalan (1989), size 24x40, 12 cells, efficacy 46.26



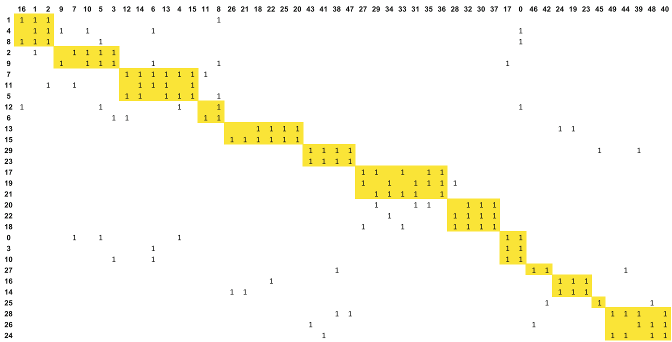
28. McCormick et al. (1972), size 27x27, 5 cells, efficacy 54.82



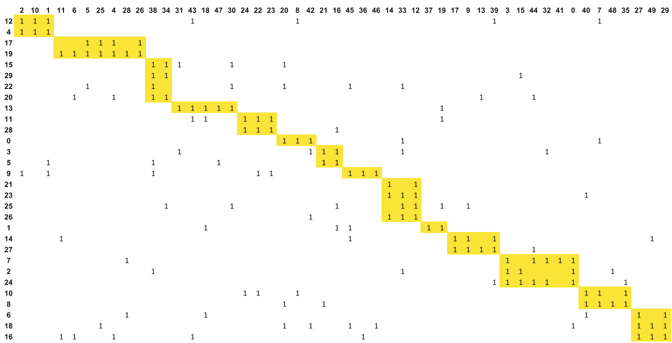
29. Carrie (1973), size 28x46, 11 cells, efficacy 47.23



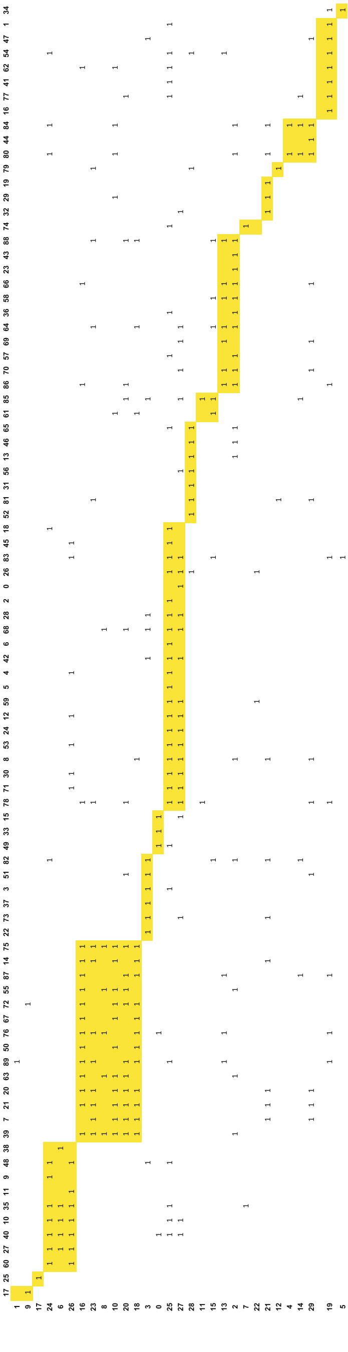
30. Kumar and Vannelli (1987), size 30x41, 15 cells, efficacy 62.77



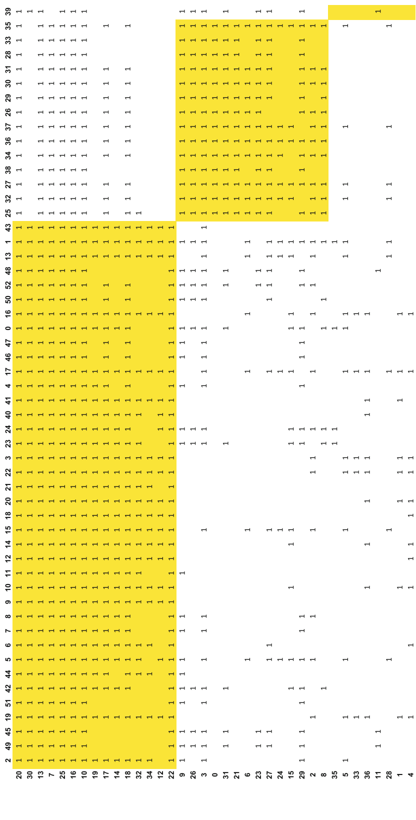
31. Stanfel (1985), size 30x50, 13 cells, efficacy 59.77



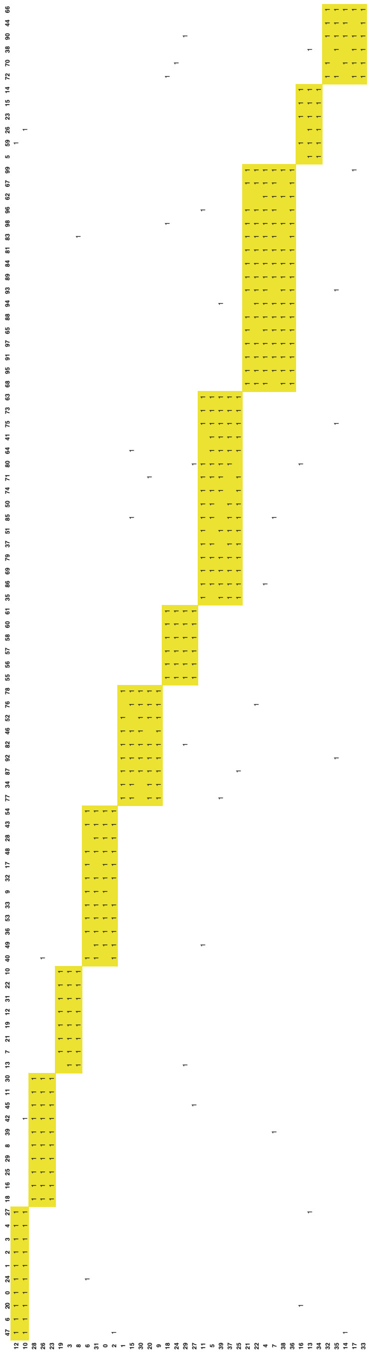
32. Stanfel (1985), size 30x50, 14 cells, efficacy 50.83



33. King and Nakornchai (1982), size 30x90, 17 cells, efficacy 48.01



34. McCormick et al. (1972), size 37x53, 3 cells, efficacy 60.50



35. Chandrasekharan and Rajagopalan (1987), size 40x100, efficacy 84.03

References

1. Askin, R.G., Chiu, K.S.: A graph partitioning procedure for machine assignment and cell formation in group technology. Int. J. Prod. Res. **28**(8), 1555–1572 (1990)
2. Askin, R.G., Subramanian, S.: A cost-based heuristic for group technology configuration. Int. J. Prod. Res. **25**, 101–113 (1987)
3. Ballakur, A., Steudel, H.J.: A within cell utilization based heuristic for designing cellular manufacturing systems. Int. J. Prod. Res. **25**, 639–655 (1987)
4. Boctor, F.: A linear formulation of the machine-part cell formation problem. Int. J. Prod. Res. **29**(2), 343–356 (1991)
5. Boe, W., Cheng, C.H.: A close neighbor algorithm for designing cellular manufacturing systems. Int. J. Prod. Res. **29**(10), 2097–2116 (1991)
6. Burbidge, J.L.: The new approach to production. Prod. Eng. **40**, 3–19 (1961)
7. Burbidge, J.L.: The Introduction of Group Technology. Wiley, New York (1975)
8. Carrie, S.: Numerical taxonomy applied to group technology and plant layout. Int. J. Prod. Res. **11**, 399–416 (1973)
9. Chan, H.M., Milner, D.A.: Direct clustering algorithm for group formation in cellular manufacture. J. Manuf. Syst. **1**(1), 64–76 (1982)
10. Chandrasekaran, M.P., Rajagopalan, R.: An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. Int. J. Prod. Res. **24**, 451–464 (1986)
11. Chandrasekaran, M.P., Rajagopalan, R.: MODROC: An extension of rank order clustering of group technology. Int. J. Prod. Res. **24**(5), 1221–1233 (1986)
12. Chandrasekharan, M.P., Rajagopalan, R.: ZODIAC—An algorithm for concurrent formation of part families and machine cells. Int. J. Prod. Res. **25**(6), 835–850 (1987)
13. Chandrasekharan, M.P., Rajagopalan, R.: Groupability: Analysis of the properties of binary data matrices for group technology. Int. J. Prod. Res. **27**(6), 1,035–1,052 (1989)
14. Cheng, C.H., Gupta, Y.P., Lee, W.H., Wong, K.F.: A TSP-based heuristic for forming machine groups and part families. Int. J. Prod. Res. **36**(5), 1325–1337 (1998)
15. Dimopoulos, C., Mort, N.: A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. Int. J. Prod. Res. **39**(1), 1–19 (2001)
16. Goldengorin, B., Krushinsky, D., Slomp, J.: Flexible PMP approach for large size cell formation, Oper. Res. **60**(5), 1157–1166 (2012)
17. Gonçalves, J.F., Resende, M.G.C.: An evolutionary algorithm for manufacturing cell formation. Comput. Ind. Eng. **47**, 247–273 (2004)
18. Hsu, C.P.: Similarity coefficient approaches to machine-component cell formation in cellular manufacturing: A comparative study. PhD Thesis, Department of Industrial and Manufacturing Engineering, University of Wisconsin Milwaukee (1990)
19. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing **38**(4), 325–340 (1987)
20. King, J.R.: Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. Int. J. Prod. Res. **18**(2), 213–232 (1980)
21. King, J.R., Nakornchai, V.: Machine-component group formation in group technology: Review and extension. Int. J. Prod. Res. **20**(2), 117–133 (1982)
22. Krushinsky, D., Goldengorin, B.: An exact model for cell formation in group technology. Comp. Manag. Sci. (2012). DOI 10.1007/s10287-012-0146-2, available at http://www.springerlink.com/content/ug2l55m46t554564/fulltext.pdf within open access at Springerlink.com
23. Kumar, K.R., Chandrasekharan, M.P.: 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 (1990)
24. Kumar K.R., Kusiak A., Vannelli A.: Grouping of parts and components in flexible manufacturing systems. Eur. J. Oper. Res. **24**, 387–97 (1986)

25. Kumar, K.R., Vannelli, A.: Strategic subcontracting for efficient disaggregated manufacturing. Int. J. Prod. Res. **25**(12), 1715–1728 (1987)
26. Kusiak, A.: The generalized group technology concept. Int. J. Prod. Res. **25**(4), 561–569 (1987)
27. Kusiak, A., Chow, W.: Efficient solving of the group technology problem. J. Manuf. Syst. **6**(2), 117–124 (1987)
28. Kusiak, A., Cho, M.: Similarity coefficient algorithm for solving the group technology problem. Int. J. Prod. Res. **30**, 2633–2646 (1992)
29. Lei, D., Wu, Z.: Tabu search for multiple-criteria manufacturing cell design. Int. J. Adv. Manuf. Tech. **28**, 950–956 (2006)
30. Liang, M., Zolfaghari, S.: Machine cell formation considering processing times and machine capacities: an ortho-synapse Hopfield neural network approach. J. Intell. Manuf. **10**, 437–447 (1999)
31. McAuley, J.: Machine grouping for efficient production. Prod. Engineer. **51**(2), 53–57 (1972)
32. McCormick, W.T., Schweitzer, P.J., White, T.W.: Problem decomposition and data reorganization by a clustering technique. Oper. Res. **20**(5), 993–1009 (1972)
33. Mitrofanov, S.P.: Nauchnye Osnovy Gruppovoy Tekhnologii, p. 435. Lenizdat, Leningrad, Russia (1959) (in Russian)
34. Mitrofanov, S.P.: The Scientific Principles of Group Technology. Boston Spa, Yorkshire: National Lending Library Translation (1966) Translation of Mitrofanov (1959)
35. Mosier, C.T.: An experiment investigating the application of clustering procedures and similarity coefficients to the GT machine cell formation problem. Int. J. Prod. Res. **27**(10), 1811–1835 (1989)
36. Mosier, C.T., Taube, L.: The facets of group technology and their impact on implementation. OMEGA **13**(6), 381–391 (1985)
37. Mosier, C.T., Taube, L.: Weighted similarity measure heuristics for the group technology machine clustering problem. OMEGA **13**(6), 577–583 (1985)
38. Ng, S.: Worst-case analysis of an algorithm for cellular manufacturing. Eur. J. Oper. Res. **69**(3), 384–398 (1993)
39. Ng, S.: On the characterization and measure of machine cells in group technology. Oper. Res. **44**(5), 735–744 (1996)
40. Onwubolu, G.C., Mutingi, M.: A genetic algorithm approach to cellular manufacturing systems. Comput. Ind. Eng. **39**(1–2), 125–144 (2001)
41. Rajagopalan, R., Batra, J.L.: Design of cellular production systems: a graph-theoretic approach. Int. J. Prod. Res. **13**(6), 567–579 (1975)
42. Seifoddini, H.: Single linkage versus average linkage clustering in machine cells formation applications. Comput. Ind. Eng. **16**(3), 419–426 (1989)
43. Seifoddini, H., Wolfe, P.M.: Application of the similarity coefficient method in group technology. IIE Trans. **18**(3), 271–277 (1986)
44. Shtub, A.: Modelling group technology cell formation as a generalized assignment problem. Int. J. Prod. Res. **27**(5), 775–782 (1989)
45. Srinivasan, G.: A clustering algorithm for machine cell formation in group technology using minimum spanning trees. Int. J. Prod. Res. **32**, 2149–2158 (1994)
46. Srinivasan, G., Narendran, T.T.: GRAFICS-A nonhierarchical clustering-algorithm for group technology. Int. J. Prod. Res. **29**(3), 463–478 (1991)
47. Srinivasan G, Narendran TT, Mahadevan B. An assignment model for the part-families problem in group technology. Int. J. Prod. Res. **28**(1), 145–152 (1990)
48. Stanfel, L.: Machine clustering for economic production. Eng. Cost. Prod. Econ. **9**, 73–81 (1985)
49. Waghodekar, P.H., Sahu, S.: Machine-component cell formation in group technology MACE. Int. J. Prod. Res. **22**, 937–948 (1984)
50. Won, Y., Lee, K.C.: Modified p-median approach for efficient GT cell formation. Comput. Ind. Eng. **46**, 495–510 (2004)
51. Wu, X., Chao-Hsien, C., Wang, Y., Yan, W.: A genetic algorithm for cellular manufacturing design and layout. Eur. J. Oper. Res. **181**, 156–167 (2007)

52. Xambre, A.R., Vilarinho, P.M.: A simulated annealing approach for manufacturing cell formation with multiple identical machines. Eur. J. Oper. Res. **151**, 434–446 (2003)
53. Yang, M.-S., Yang, J.-H.: Machine-part cell formation in group technology using a modified ART1 method. Eur. J. Oper.Res. **188**(1), 140–152 (2008)