Heuristic for Maximizing Grouping Efficiency in the Cell Formation Problem

Ilya Bychkov, Mikhail Batsyn and Panos M. Pardalos

Abstract In our paper, we consider the Cell Formation Problem in Group Technology with grouping efficiency as an objective function. We present a heuristic approach for obtaining high-quality solutions of the CFP. The suggested heuristic applies an improvement procedure to obtain solutions with high grouping efficiency. This procedure is repeated many times for randomly generated cell configurations. Our computational experiments are performed for popular benchmark instances taken from the literature with sizes from 10×20 to 50×150. Better solutions unknown before are found for 23 instances of the 24 considered. The preliminary results for this paper are available in Bychkov et al. (Models, algorithms, and technologies for network analysis, Springer, NY, vol. 59, pp. 43–69, 2013, [7]).

1 Introduction

Flanders [15] was the first who formulated the main ideas of the group technology. The notion of the Group Technology was introduced in Russia by [30], though his work was translated to English only in 1966 [31]. One of the main problems stated by the Group Technology is the optimal formation of manufacturing cells, i.e., grouping of machines and parts into cells such that for every machine in a cell the number of the parts from this cell processed by this machine is maximized and the number of the parts from other cells processed by this machine is minimized. In other words,

I. Bychkov (✉) · M. Batsyn · P.M. Pardalos
Laboratory of Algorithms and Technologies for Network Analysis,
National Research University Higher School of Economics,
136 Rodionova Street, Nizhny Novgorod 603093, Russia
e-mail: ibychkov@hse.ru

M. Batsyn
e-mail: mbatsyn@hse.ru

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

V.A. Kalyagin et al. (eds.), *Models, Algorithms, and Technologies for Network Analysis*, Springer Proceedings in Mathematics & Statistics 197,
DOI 10.1007/978-3-319-56829-4_2

the intra-cell loading of machines is maximized and simultaneously the inter-cell movement of parts is minimized. This problem is called the Cell Formation Problem (CFP). Burbidge [5] suggested his Product Flow Analysis (PFA) approach for the CFP, and later popularized the Group Technology and the CFP in his book [6].

The CFP is NP-hard since it can be reduced to the clustering problem [16]. That is why there is a great number of heuristic approaches for solving CFP and almost no exact ones. The first algorithms for solving the CFP were different clustering techniques. Array-based clustering methods find rows and columns permutations of the machine-part matrix in order to form a block-diagonal structure. These methods include: Bond Energy Algorithm (BEA) of [29], Rank Order Clustering (ROC) algorithm by [20], its improved version ROC2 by [21], Direct Clustering Algorithm (DCA) of [12], Modified Rank Order Clustering (MODROC) algorithm by [9], the Close Neighbor Algorithm (CAN) by [4]. Hierarchical clustering methods at first form several big cells, 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 [28], Average Linkage [39], and Complete Linkage [32] algorithms. Nonhierarchical clustering methods are iterative approaches which start from some initial partition and improve it iteratively. The two most successful are GRAFICS algorithm by [41] and ZODIAC algorithm by [10]. A number of works considered the CFP as a graph partitioning problem, where machines are vertices of a graph. [37] used clique partitioning of the machines graph. Askin and Chiu [2] implemented a heuristic partitioning algorithm to solve CFP. Ng [35, 36] suggested an algorithm based on the minimum spanning tree problem. Mathematical programming approaches are also very popular for the CFP. Since the objective function of the CFP is rather complicated from the mathematical programming point of view most of the researchers use some approximation model which is then solved exactly for small instances and heuristically for large. [25] formulated CFP via p-median model and solved several small size CFP instances, [40] used Generalized Assignment Problem as an approximation model, [44] proposed a simplified p-median model for solving large CFP instances, [22] applied minimum k-cut problem to the CFP, [17] used p-median approximation model and solved it exactly by means of their pseudo-boolean approach including large CFP instances up to 50×150 instance. A number of meta-heuristics have been applied recently to the CFP. Most of these approaches can be related to genetic, simulated annealing, Tabu search, and neural networks algorithms. Among them are works such as: [18, 26, 27, 45–47].

Our heuristic algorithm is based on sequential improvements of the solution. We modify the cell configuration by enlarging one cell and reducing another. The basic procedure of the algorithm has the following steps:

1. Generate a random cell configuration.
2. Improve the initial solution moving one row or column from one cell to another until the grouping efficiency is increasing.
3. Repeat steps 1–2 a predefined number of times (we use 2000 times for computational experiments in this paper).

The paper is organized as follows. In the next section, we provide the Cell Formation Problem formulation. In Sect. 3 we present our improvement heuristic that allows us to get good solutions by iterative modifications of cells which lead to increasing of the objective function. In Sect. 4 we report our computational results and Sect. 5 concludes the paper with a short summary.

2 The Cell Formation Problem

The CFP consists in an optimal grouping of the given machines and parts into cells. The input for this problem is 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 grouping of rows (machines) and columns (parts) into these cells such that the number of zeros inside the chosen cells (voids) and the number of ones outside these cells (exceptions) are minimized. A concrete combination of rectangular cells in a solution (diagonal blocks in the machine-part matrix) we will call a cells configuration. Since it is usually not possible to minimize these two values simultaneously there have appeared a number of compound criteria trying to join it into one objective function. Some of them are presented below.

For example, we are given the machine-part matrix [43] shown in Table 1. Two different solutions for this CFP are shown in Tables 2 and 3. The left solution is better because it has less voids (3 against 4) and exceptions (4 against 5) than the right one. But one of its cells is a singleton—a cell which has less than two machines or parts.

Table 1 Machine-part 5 × 7 matrix from [43]

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

Table 2 Solution with 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 3 Solution without singletons

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

In some CFP formulations singletons are not allowed, so in this case this solution is not feasible. In this paper, we consider both the cases (with allowed singletons and with not allowed) and when there is a solution with singletons found by the suggested heuristic better than without singletons we present both the solutions.

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 [11]:

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

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 intra-cell 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 zeroes out of cells to the total number of elements out of cells).
q—a coefficient ($0 \le q \le 1$) reflecting the weights of the machine loading and the inter-cell movement in the objective function. It is usually taken equal to $\frac{1}{2}$, which means that it is equally important to maximize the machine loading and minimize the inter-cell movement.
n_1—a number of ones in the machine-part matrix,
n_0—a number of zeroes 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 zeroes inside the cells,
n_0^{out}—a number of zeroes outside the cells.

2. Grouping efficacy suggested by [23]:

$$\tau = \frac{n_1 - n_1^{out}}{n_1 + n_0^{in}} = \frac{n_1^{in}}{n_1 + n_0^{in}} \tag{2}$$

3. Group Capability Index (GCI) suggested by [19]:

$$GCI = 1 - \frac{n_1^{out}}{n_1} = \frac{n_1 - n_1^{out}}{n_1} \tag{3}$$

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

$$E + V = n_1^{out} + n_0^{in} \tag{4}$$

The values of these objective functions for the solutions in Tables 2 and 3 are shown below.

$$\eta = \frac{1}{2}\cdot\frac{16}{19} + \frac{1}{2}\cdot\frac{12}{16} \approx 79.60\% \qquad \eta = \frac{1}{2}\cdot\frac{15}{19} + \frac{1}{2}\cdot\frac{11}{16} \approx 73.85\%$$

$$\tau = \frac{20-4}{20+3} \approx 69.57\% \qquad \tau = \frac{20-5}{20+4} \approx 62.50\%$$

$$GCI = \frac{20-4}{20} \approx 80.00\% \qquad GCI = \frac{20-5}{20} \approx 75.00\%$$

$$E + V = 4 + 3 = 7 \qquad E + V = 5 + 4 = 9$$

In this paper, we use the grouping efficiency measure and compare our computational results with the results of [17, 47].

The mathematical programming model of the CFP with the grouping efficiency objective function can be described using boolean variables x_{ik} and y_{jk}. Variable x_{ik} takes value 1 if machine i belongs to cell k and takes value 0 otherwise. Similarly variable y_{jk} takes value 1 if part j belongs to cell k and takes value 0 otherwise. Machines index i takes values from 1 to m and parts index j - from 1 to p. Cells index k takes values from 1 to $c = \min(m, p)$ because every cell should contain at least one machine and one part, and so the number of cells cannot be greater than m and p. Note, that if a CFP solution has n cells then for k from $n + 1$ to c all variables x_{ik}, y_{jk} will be zero in this model. So, we can consider that the CFP solution always has c cells, but some of them can be empty. The mathematical programming formulation is as follows:

$$\max\left(\frac{n_1^{in}}{2n^{in}} + \frac{n_0^{out}}{2n^{out}}\right) \tag{5}$$

where

$$n^{in} = \sum_{k=1}^{c}\sum_{i=1}^{m}\sum_{j=1}^{p} x_{ik}y_{jk}, \quad n^{out} = mp - n^{in}$$

$$n_1^{in} = \sum_{k=1}^{c}\sum_{i=1}^{m}\sum_{j=1}^{p} a_{ij}x_{ik}y_{jk}, \quad n_0^{out} = n_0 - (n^{in} - n_1^{in})$$

subject to

$$\sum_{k=1}^{c} x_{ik} = 1 \quad \forall i \in 1, ..., m \tag{6}$$

$$\sum_{k=1}^{c} y_{jk} = 1 \quad \forall j \in 1, ..., p \tag{7}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{p} x_{ik}y_{jk} \geq \sum_{i=1}^{m} x_{ik} \quad \forall k \in 1, ..., c \tag{8}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{p} x_{ik}y_{jk} \geq \sum_{j=1}^{p} y_{jk} \quad \forall k \in 1, ..., c \tag{9}$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in 1, ..., m \tag{10}$$

$$y_{jk} \in \{0, 1\} \quad \forall j \in 1, ..., p \tag{11}$$

The objective function (5) is the grouping efficiency in this model. Constraints (6) and (7) impose that every machine and every part belongs to some cell. Constraints (8) and (9) guarantee that every nonempty cell contains at least one machine and one part. Note that if singleton cells are not allowed then the right sides of inequalities (8) and (9) should have a coefficient of 2. All these constraints can be linearized in a standard way, but the objective function will still be fractional. That is why the exact solution of this problem presents considerable difficulties.

A cells configuration in the mathematical model is described by the number of machines m_k and parts p_k in every cell k.

$$m_k = \sum_{i=1}^{m} x_{ik}, \quad p_k = \sum_{j=1}^{p} y_{jk}$$

It is easy to see that when a cells configuration is fixed all the optimization criteria (1)–(4) become equivalent (Proposition 1).

Proposition 1 *If a cells configuration is fixed then objective functions (1)–(4): η, τ, GCI, $E+V$ become equivalent and reach the optimal value on the same solutions.*

Proof When a cells configuration is fixed the following values are constant: m_k, p_k. The values of n_1 and n_0 are always constant. The values of n^{in} and n^{out} are constant since $n^{in} = \sum_{k=1}^{c} m_k p_k$ and $n^{out} = mp - n^{in}$. So, if we maximize the number of ones inside the cells n_1^{in} then simultaneously $n_0^{in} = n^{in} - n_1^{in}$ 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 simultaneously on the same optimal solution. □

3 Algorithm Description

The main function of our heuristic is presented by algorithm 1.

Algorithm 1 Main function

```
function SOLVE( )
    FINDOPTIMALCELLRANGE(MinCells, MaxCells)
    ConfigsNumber = 2000
    AllConfigs = GENERATECONFIGS(MinCells, MaxCells, ConfigsNumber)
    return CMHEURISTIC(AllConfigs)
end function
```

First we call FINDOPTIMALCELLRANGE($MinCells$, $MaxCells$) function that returns a potentially optimal range of cells - from $MinCells$ to $MaxCells$. Then these values and $ConfigsNumber$ (the number of cell configurations to be generated) are passed to GENERATECONFIGS($MinCells$, $MaxCells$, $ConfigsNumber$) function which generates random cell configurations. The generated configurations $AllConfigs$ are passed to CMHEURISTIC($AllConfigs$) function which finds a high-quality solution for every cell configuration and then chooses the solution with the greatest efficiency value.

Algorithm 2 Procedure for finding the optimal cell range

```
function FindOptimalCellRange( MinCells, MaxCells)
    if (m > p) then
        minDimension = p
    else
        minDimension = m
    end if
    ConfigsNumber = 500
    Configs = GenerateConfigs(2, minDimension, ConfigsNumber)
    Solution = CMHeuristic(Configs)
    BestCells = GetCellsNumber(Solution)
    MinCells = BestCells - [minDimension * 0,1 ]        ▷ [ ] - integer part
    MaxCells = BestCells + [minDimension * 0,1 ]
end function
```

In function FindOptimalCellRange($MinCells$, $MaxCells$) (Algorithm 2) we look over all the possible number of cells from 2 to maximal possible number of cells which is equal to $\min(m, p)$. For every number of cells in this interval, we generate a fixed number of configurations (we use 500 in this paper) calling GenerateConfigs(2, $minDimension$, $ConfigsNumber$) and then use our CMHeuristic($Configs$) to obtain a potentially optimal number of cells. But we consider not only one number of cells but together with its 10%-neighborhood [$MinCells$, $MaxCells$].

Algorithm 3 Configurations generation

```
function GenerateConfigs(MinCells, MaxCells, ConfigsNumber)
    Configs = ∅
    for cells = MinCell, MaxCells do
        Generated = GenerateConfigs(cells, ConfigsNumber)
        Configs = Configs ∪ Generated
        return Configs
    end for
end function
```

Function GenerateConfigs($MinCells$, $MaxCells$, $ConfigsNumber$) (Algorithm 3) returns a set of randomly generated cell configurations with a number of cells ranging from $MinCells$ to $MaxCells$. We call GenerateConfigsUniform($cells$, $ConfigsNumber$) function which randomly selects with uniform distribution $ConfigsNumber$ configurations from all possible cell configurations with the specified number of cells. Note that mathematically a cell configuration with k cells can be represented as an integer partition of m and p values into sums of k summands. We form a set of configurations for every number of cells and then join them.

Algorithm 4 CMHeuristic

function CMHEURISTIC($Configs$)
  $Best = 0$
  for all $config \in Configs$ **do**
    $Solution =$ IMPROVESOLUTION($config$)
    if $Solution > Best$ **then**
      $Best = Solution$
    end if
  end for
  return $Best$
end function

Function CMHEURISTIC(Configs) (Algorithm 4) gets a set of cell configurations and for each configuration runs an improvement algorithm to obtain a good solution. A solution includes a permuted machine-part matrix, a cell configuration, and the corresponding grouping efficiency value. The function chooses the best solution and returns it.

Improvement procedure IMPROVESOLUTION($config, \eta_{current}$) (Algorithm 5) works as follows. We consider all the machines and the parts in order to know if there is a machine or a part that we can move to another cell and improve the current efficiency $\eta_{current}$. First we consider moving of every part on all other cells and compute how the efficiency value changes. Here $\eta_{part,cell}$ is the efficiency of the current solution where the part with index $part$ is moved to the cell with index $cell$. This operation is performed for all the parts and the part with the maximum increase in efficiency Δ_{parts} is chosen. Then we repeat the same operations for all the machines. Finally, we compare the best part movement and the best machine movement and choose the one with the highest efficiency. This procedure is performed until any improvement is possible and after that we get the final solution.

The main idea of IMPROVESOLUTION($config, \eta_{current}$) is illustrated on [39] instance 8×12 (Table 4). To compute the grouping efficiency for this solution, we need to know the number of ones inside cells n_1^{in}, the total number of elements inside

Table 4 [39] instance 8×12

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	0	0	0	0	0	0	0	0
2	1	0	1	1	1	1	1	0	0	1	0	0
3	0	0	1	1	1	1	1	1	1	0	0	0
4	0	0	0	0	0	1	1	1	1	1	0	0
5	0	0	0	0	0	0	1	1	1	1	0	0
6	0	0	0	0	0	0	1	1	1	0	1	0
7	0	0	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0	1	1

Algorithm 5 Solution improvement procedure

```
function ImproveSolution(config, η_current)
    η_current = GroupingEfficiency(config)
    repeat
        PartFrom = 0
        PartTo = 0
        for part = 1, partsNumber do
            for cell = 1, cellsNumber do
                if (η_part,cell > η_current) then
                    Δ_parts = (η_part,cell − η_current)
                    PartFrom = GetPartCell(part)
                    PartTo = cell
                end if
            end for
        end for
        MachineFrom = 0
        MachineTo = 0
        for machine = 1, machinesNumber do
            for cell = 1, cellsNumber do
                if (η_machine,cell > η_current) then
                    Δ_machines = (η_machine,cell − η_current)
                    MachineFrom = GetMachineCell(machine)
                    MachineTo = cell
                end if
            end for
        end for
        if Δ_parts > Δ_machines then
            MovePart(PartFrom, PartTo)
        else
            MoveMachine(MachineFrom, MachineTo)
        end if
    until Δ > 0
end function
```

cells n^{in}, the number of zeros outside cells n_0^{out}, and the number of elements outside cells n^{out}. The grouping efficiency is then calculated by the following formula:

$$\eta = q \cdot \frac{n_1^{in}}{n^{in}} + (1-q) \cdot \frac{n_0^{out}}{n^{out}} = \frac{1}{2} \cdot \frac{20}{33} + \frac{1}{2} \cdot \frac{48}{63} \approx 68.4\%$$

Looking at this solution (Table 4) we can conclude that it is possible, for example, to move part 4 from the second cell to the first one. And this way, the number of zeros inside cells decreases by 3 and the number of ones outside cells also decreases by 4. So, it is profitable to attach column 4 to the first cell as it is shown on Table 5. For the modified cells configuration we have:

$$\eta = \frac{1}{2} \cdot \frac{23}{33} + \frac{1}{2} \cdot \frac{51}{63} \approx 75.32\%$$

Table 5 Moving part 4 from cell 2 to cell 1

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	0	0	0	0	0	0	0	0
2	1	0	1	1	1	1	1	0	0	1	0	0
3	0	0	1	1	1	1	1	1	1	0	0	0
4	0	0	0	0	0	1	1	1	1	1	0	0
5	0	0	0	0	0	0	1	1	1	1	0	0
6	0	0	0	0	0	0	1	1	1	0	1	0
7	0	0	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0	1	1

Table 6 Maximal efficiency increase for each row

	1	2	3	4	5	6	7	8	9	10	11	12	
1	1	1	1	1	0	0	0	0	0	0	0	0	−6.94%
2	1	0	1	1	1	1	1	0	0	1	0	0	+1.32%
3	0	0	1	1	1	1	1	1	1	0	0	0	+7.99%
4	0	0	0	0	0	1	1	1	1	1	0	0	−0.07%
5	0	0	0	0	0	0	1	1	1	1	0	0	+0.77%
6	0	0	0	0	0	0	1	1	1	0	1	0	+0.77%
7	0	0	0	0	0	0	0	0	0	0	1	1	−4.62%
8	0	0	0	0	0	0	0	0	0	0	1	1	−4.62%

As a result the efficiency is increased almost for 7%. Computational results show that using such modifications could considerably improve the solution. The idea is to compute an increase in efficiency for each column and row when it is moved to another cell and then perform the modification corresponding to the maximal increase. For example, Table 6 shows the maximal possible increase in efficiency for every row when it is moved to another cell.

4 Computational Results

In all the experiments for determining a potentially optimal range of cells we use 500 random cell configurations for each cells number, and for obtaining the final solution we use 2000 random configurations. An Intel Core i7 machine with 2.20 GHz CPU and 8.00 Gb of memory is used in our experiments. We run our heuristic on 24 CFP benchmark instances taken from the literature. The sizes of the considered problems vary from 10×20 to 50×150. The computational results are presented in Table 7. For every instance we make 50 algorithm runs and report minimum, average, and maximum value of the grouping efficiency obtained by the suggested heuristic

Table 7 Computational results

#	Source	mxp	Efficiency value, %					Time, sec	Cells
			Bhatnagar and Saddikuti	Goldengorin et al.	Our				
					Min	Avg	Max		
1	Sandbothe [38]	10×20	95.40	95.93[a]	95.66	95.66	**95.66**	0.36	7
2	Ahi et al. [1]	20×20	92.62	93.85	95.99	95.99	**95.99**	0.62	9
3	Mosier and Taube [33]	20×20	85.63	88.71	90.11	90.16	**90.22**	0.88	9
4	Boe and Cheng [4]	20×35	88.31	88.05	93.34	93.47	**93.55**	1.62	10
5	Carrie [8]	20×35	90.76	95.64	95.43	95.78	**95.79**	1.54	10
6	Ahi et al. [1]	20×51	87.86	94.11	95.36	95.4	**95.45**	3.1	12
7	Chandrasekharan et al. [11]	20×40	98.82	100.00	100	100	100	1.8	7
8	Chandrasekharan et al. [11]	20×40	95.33	97.48	97.7	97.75	**97.76**	2.42	12
9	Chandrasekharan et al. [11]	20×40	93.78	96.36	96.84	96.88	**96.89**	2.56	12
10	Chandrasekharan et al. [11]	20×40	87.92	94.32	96.11	96.16	**96.21**	3.3	15
11	Chandrasekharan et al. [11]	20×40	84.95	94.21	95.94	96.03	**96.1**	2.84	15
12	Chandrasekharan et al. [11]	20×40	85.06	92.32	95.85	95.9	**95.95**	2.76	15
13	Nair and Narendran [34]	20×40	96.44	97.39	97.78	97.78	**97.78**	2.12	10
14	Nair and Narendran [34]	20×40	92.35	95.74	97.4	97.4	**97.4**	2.2	14
15	Nair and Narendran [34]	20×40	93.25	95.70	95.81	96.03	**96.17**	2.48	12
16	Nair and Narendran [34]	20×40	91.11	96.40	96.98	96.98	**96.98**	2.78	14
17	Ahi et al. [1]	25×40	91.09	95.52	96.48	96.48	**96.48**	2.58	14
18	Yang and Yang [47]	28×35	93.43	93.82	94.81	94.85	**94.86**	2.46	10
19	Kumar and Vannelli [24]	30×41	90.66	97.22	97.38	97.53	**97.62**	3.54	18
20	Stanfel [42]	30×50	88.17	96.48	96.77	96.83	**96.9**	5.02	18

(continued)

Table 7 (continued)

#	Source	mxp	Efficiency value, %					Time, sec	Cells
			Bhatnagar and Saddikuti	Goldengorin et al.	Our				
					Min	Avg	Max		
21	King and Nakornchai [21]	30×90	83.18	94.62	95.37	95.84	**96.27**	13.1	25
22	Chandrasekharan and Rajagopalan [10]	40×100	94.75	95.91	98.06	98.1	**98.13**	16.88	17
23	Yang and Yang [47]	46×105	90.98	95.20	96.1	96.18	**96.29**	23.9	18
24	Liang and Zolfaghari [27]	50×150	93.05	92.92	96.08	96.17	**96.27**	51.66	24

[a]This solution has a mistake

over these 50 runs. We compare our results with the best known values taken from [3, 17]. We have found better solutions unknown before for 23 instances of the 24 considered. For CFP instance 6, we have found the same optimal solution with 100% of grouping efficiency as in [17]. For CFP instance 1 the solution of [17] has some mistake. For this instance having a small size of 10×20 it can be proved that our solution is the global optimum applying an exact approach [14] for the grouping efficiency objective and all the possible number of cells from 1 to 10.

5 Concluding Remarks

In this paper, we present a new heuristic algorithm for solving the CFP. The high quality of the solutions is achieved due to the enumeration of different numbers of cells and different cell configurations and applying our improvement procedure. Since the suggested heuristic works fast (the solution for one cell configuration is achieved in several milliseconds for any instance from 10×20 to 50×150), we apply it for thousands of different configurations. Thus a big variety of good solutions is covered by the algorithm and the best of them has high grouping efficiency.

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

References

1. Ahi, A., Mir, B., Aryanezhad, B., Ashtiani, A.M.: 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 (2009)
2. 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)
3. Bhatnagar, R., Saddikuti, V.: Models for cellular manufacturing systems design: matching processing requirements and operator capabilities. J. Opl. Res. Soc. **61**, 827–839 (2010)
4. Boe, W., Cheng, C.H.: A close neighbor algorithm for designing cellular manufacturing systems. Int. J. Prod. Res. **29**(10), 2097–2116 (1991)
5. Burbidge, J.L.: The new approach to production. Prod. Eng. (1961). (Dec. 3–19)
6. Burbidge, J.L.: The Introduction of Group Technology. Wiley, New York (1975)
7. Bychkov, I., Batsyn, M., Sukhov, P.: Ch. 4., Heuristic algorithm for the cell formation problem. In: Goldengorin, B.I., Kalyagin, V.A., Pardalos, P.M. (eds.) Models, Algorithms, and Technologies for Network Analysis, vol. 59, pp. 43–69. Springer, NY (2013)
8. Carrie, S.: Numerical taxonomy applied to group technology and plant layout. Int. J. Prod. Res. **11**, 399–416 (1973)
9. Chandrasekaran, M.P., Rajagopalan, R.: MODROC: an extension of rank order clustering of group technology. Int. J. Prod. Res. **24**(5), 1221–1233 (1986)
10. 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)

11. Chandrasekharan, M.P., Rajagopalan, R.: Groupability: analysis of the properties of binary data matrices for group technology. Int. J. Prod. Res. **27**(6), 1035–1052 (1989)
12. Chan, H.M., Milner, D.A.: Direct clustering algorithm for group formation in cellular manufacture. J. Manuf. Syst. **1**(1), 64–76 (1982)
13. 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)
14. Elbenani, B., Ferland, J.A.: Manufacturing Cell Formation by State-Space Search, pp. 14. Montreal, Quebec. CIRRELT-2012-07 (2012)
15. Flanders, R.E.: Design manufacture and production control of a standard machine. Trans. ASME **46**, 691–738 (1925)
16. Ghosh, S., Mahanti, A., Nagi, R., Nau, D.S.: Cell formation problem solved exactly with the dinkelbach algorithm. Ann. Oper. Res. **65**(1), 35–54 (1996)
17. Goldengorin, B., Krushinsky, D., Slomp, J.: Flexible PMP approach for large size cell formation. Oper. Res. **60**(5), 1157–1166 (2012)
18. Goncalves, J.F., Resende, M.G.C.: An evolutionary algorithm for manufacturing cell formation. Comput. Ind. Eng. **47**, 247–273 (2004)
19. Hsu, C.P.: Similarity coefficient approaches to machine-component cell formation in cellular manufacturing: a comparative study. Ph.D. Thesis, Department of Industrial and Manufacturing Engineering, University of Wisconsin Milwaukee (1990)
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. CMS **9**, 323–338 (2012). doi:10.1007/s10287-012-0146-2
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., Vannelli, A.: Strategic subcontracting for efficient disaggregated manufacturing. Int. J. Prod. Res. **25**(12), 1715–1728 (1987)
25. Kusiak, A.: The generalized group technology concept. Int. J. Prod. Res. **25**(4), 561–569 (1987)
26. Lei, D., Wu, Z.: Tabu search for multiple-criteria manufacturing cell design. Int. J. Adv. Manuf. Technol. **28**, 950–956 (2006)
27. Liang, M., Zolfaghari, S.: Machine cell formation considering processing times and machine capacities: an ortho-synapse Hopfield neural netowrk approach. J. Intell. Manuf. **10**, 437–447 (1999)
28. McAuley, J.: Machine grouping for efficient production. Prod. Eng. Res. Dev. **51**(2), 53–57 (1972)
29. 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)
30. Mitrofanov, S.P.: Nauchnye osnovy gruppovoy tekhnologii. Lenizdat, Leningrad, Russia, 435 pp (1959) (in Russian)
31. Mitrofanov, S.P.: The Scientific Principles of Group Technology. Boston Spa. National Lending Library Translation, Yorkshire (1966) (Translation of Mitrofanov (1959))
32. 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)
33. Mosier, C.T., Taube, L.: Weighted similarity measure heuristics for the group technology machine clustering problem. OMEGA **13**(6), 577–583 (1985)
34. Nair, G.J.K., Narendran, T.T.: Grouping index: a new quantitative criterion for goodness of block-diagonal forms in group technology. Int. J. Prod. Res. **34**(10), 2767–2782 (1996)
35. Ng, S.: Worst-case analysis of an algorithm for cellular manufacturing. Eur. J. Oper. Res. **69**(3), 384–398 (1993)

36. Ng, S.: On the characterization and measure of machine cells in group technology. Oper. Res. **44**(5), 735–744 (1996)
37. Rajagopalan, R., Batra, J.L.: Design of cellular production systems: a graph-theoretic approach. Int. J. Prod. Res. **13**(6), 567–579 (1975)
38. Sandbothe, R.A.: Two observations on the grouping efficacy measure for goodness of block diagonal forms. Int. J. Prod. Res. **36**, 3217–3222 (1998)
39. Seifoddini, H., Wolfe, P.M.: Application of the similarity coefficient method in group technology. IIE Trans. **18**(3), 271–277 (1986)
40. Shtub, A.: Modelling group technology cell formation as a generalized assignment problem. Int. J. Prod. Res. **27**(5), 775–782 (1989)
41. Srinivasan, G., Narendran, T.T.: GRAFICS-A nonhierarchical clustering-algorithm for group technology. Int. J. Prod. Res. **29**(3), 463–478 (1991)
42. Stanfel, L.: Machine clustering for economic production. Eng. Costs Prod. Econ. **9**, 73–81 (1985)
43. Waghodekar, P.H., Sahu, S.: Machine-component cell formation in group technology MACE. Int. J. Prod. Res. **22**, 937–948 (1984)
44. Won, Y., Lee, K.C.: Modified p-median approach for efficient GT cell formation. Comput. Ind. Eng. **46**, 495–510 (2004)
45. 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)
46. 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)
47. 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)