



# An immune system based algorithm for cell formation problem

Berna H. Ulutas<sup>1</sup>

Received: 20 November 2017 / Accepted: 22 February 2018 / Published online: 3 March 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

Technological developments enable the design and manufacturing of products tailored to individual consumers. Cellular Manufacturing Systems (CMS) can be considered as to ease flexibility, to reduce setup time, throughput time, work-in-process inventories, and material handling costs. Cell formation problem (CFP) that is one of the critical CMS design problems is the assignment of parts and machines to specific cells based on their similarity. This study introduces a Clonal Selection Algorithm (CSA) with a novel encoding structure that is efficient to solve real-sized problems. Unlike the methods in literature that define the number of cells as a constant number, this algorithm is significant because it can obtain the optimum number of cell to generate best efficacy value. Proposed CSA is tested by using 67 (35 well-known and 32 less-known) test problems. CSA obtains the same 63 best-known optimal solutions, provides solutions for the 3 of the well-known test problem and a new solution for the largest test problem (50 machine 150 part) that was not possible to be solved by the mixed integer linear programming model due to the high computational complexity. Final CSA grouping results are illustrated with figures to attract attention to the singleton and residual cells.

**Keywords** Artificial immune systems · Cell formation problem · Clonal selection algorithm · Group technology

## Introduction

The advances in internet and digital services have impacted the rise in automation. Therefore, today's manufacturing systems have a tendency to shift from standardization through mass manufacturing, towards personalization by utilizing the concepts in inclusive manufacturing. This also has a reflection to the facility layouts because the product demands are likely to become mid-volume and mid-variety mixes due to the global competition and short product life cycles. Classical job shop and flow shop configurations cannot handle such needs efficiently and flexibly. Cellular Manufacturing (CM) is an implementation of the GT concept for designing manufacturing systems that is known to focus on similarities between parts, products and processes to achieve efficiencies (Hyer and Wemmerlov 1984). The design of CMS involve interrelated sub problems such as, machine grouping and part family formation (cell formation), intra-cell layout (machine layout) and inter-cell layout (cell layout) (Arvindh and Irani 1994). The optimization of the cell formation problem (CFP)

is proved to be a nondeterministic polynomial (NP) complete problem (Dimopoulos and Zalzala 2000). Manufacturing cells are clusters of dissimilar machines placed in close proximity that are dedicated to the manufacture of families of parts (Wemmerlov and Hyer 1989). Kamel et al. (1994) propose an algorithm for assigning parts to machines which utilizes the types of operations required by parts and applies group technology principles. Dimopoulos and Zalzala (2000), Goncalves and Resende (2004) state that traditional optimization methods can not find optimal solutions to large sized CFPs within a reasonable amount of time.

A comprehensive review, comparison and evaluation of methodologies (1997–2008) is summarized in Papaionnaou and Wilson (2010). Cell formation methods are mainly grouped as mathematical programming, heuristics, meta-heuristics, hybrid metaheuristics, and artificial intelligence approaches. Several methodologies are used to deal with cellular manufacturing problems. Since the aim of the study is not a comprehensive survey, a number of studies related with CFP literature that mainly use same test problem data in this paper can be listed as follows:

✉ Berna H. Ulutas  
bhaktan@ogu.edu.tr

<sup>1</sup> Department of Industrial Engineering, Eskisehir Osmangazi University, 26480 Eskisehir, Turkey

• *ZODIAC method* Chandrasekharan and Rajagopalan (1987),

- *GRAFICS method* Srinivasan et al. (1990),
- *Clustering* Srinivasan (1994),
- *Genetic Algorithm (GA)* Joines et al. (1996), Cheng et al. (1998), Dimopoulos and Mort (2001), Onwubalu and Mutungi (2001), Brown and Sumichrast (2001), Zolfaghari and Liang (2002), Islier (2001), Yasuda et al. (2005), Tunnuwij and Hicks (2009), Mahdavi et al. (2009), Arkat et al. (2011),
- *Neural Networks (NN)* Malavi and Ramachandran (1991), Lee et al. (1992), Moon (1992), Moon and Chi (1992), Onwubalu (1999),
- *Simulated Annealing (SA)* Zolfaghari and Liang (2002), Islier (2005), James et al. (2007),
- *Tabu Search (TS)* (Zolfaghari and Liang (2002), Islier (2005),
- *Evolutionary algorithm (EA)* Goncalves and Resende (2004),
- *Ant system algorithm, Ant Colony Optimization (ACO)* Islier (2005), Xiangyong et al. (2010), Farahani and Hosseini (2011).

Ulutas and Islier (2007) introduce CSA for a typical facility layout problem with the equal area assumption and study the effect of parameters for 4 small and 4 large sized test problems. Results illustrate the superiority of the algorithm for large sized problems. Ulutas and Islier (2009a) focus on 8 dynamic facility layout test problems and proved that CSA outperforms 88% of the large sized problems. Ulutas and Kulturel-Konak (2012) utilize CSA to solve unequal area facility layout problem and test the performance of the algorithm by use of 25 test problems. CSA obtained better results for the 95.65% of the test problems with the best-so-far flexible bay structure results. Ulutas and Kulturel-Konak (2013) focus on unequal area facility layout problem with flexible bay structure and state that the inverse mutation followed by pairwise mutation has a potential to obtain better results within short computational times.

Ulutas and Islier (2009b) is the first study to introduce a CSA for CFP. 20 test problems are used to compare CSA results with other nature based methods (such as GA, SA, TS, artificial ant systems and artificial neural networks). Computational results prove that CSA does not require long computing times and provides the same best solutions in current literature. On the other hand, it is pointed that CSA generates better results compared to ant system algorithms and GA. Ulutas and Sarac (2009) consider alternative routes for a CFP and introduces three-segment antibody representation for CSA. Algorithm is tested on 4 test problems and it is observed that best known solutions for small sized problems are obtained in short computational times. Ulutas (2015) obtains best results for 20 test problems in the literature by CSA. Further, results for a 16 machine 43 part test problem is studied in detail to attract attention on the singletons and

state that CFP results should be assessed not only based on efficacy values but also consider the number of cells.

Main contribution of this paper is the definition of a novel CSA encoding for CFP that is capable to handle realistic problem sizes. The basic interactive software developed for the test problems has a potential to be used in industrial applications. It is known that, due to the combinatorial complexity of the problem, researchers may consider fewer constraints and variables. CFP results generated by use of mathematical programming are provided only based on the performance measure (i.e., efficacy, efficiency value) and final grouping is not represented. In contrast, visual representation of the CSA results is also a contribution for researchers to compare their results and for managers to support their decision in a real life problem.

Papaionnaou and Wilson (2010) also state that there is still a need for objectively comparing the results on benchmark problems. Several researchers have used well-known test problems for cell formation computational experiments from Goncalves and Resende (2004). Unfortunately some of the tests have inconsistencies with the original reference and a number of the best solutions in the literature are stated in two digits where most are stated in four digits that lead to misleading interpretation. Therefore, this paper aims to consider the test problems in a broader view and compile 35 well-known test problems with accessible results in the literature. Further, additional 32 less-known test problems named as Data set B from Bychkov and Batsyn (2018) is considered.

The paper is structured as follows: Section 2 gives brief CSA steps and explains the proposed CFP encoding. Test problems from the literature and results obtained by CSA are reported in Sect. 3. Conclusions and future study directions are summarized in the last section.

## Clonal selection algorithm for cell formation problem

Metaheuristics are mainly developed to solve combinatorial optimization problems. Based on the literature review, trajectory methods (SA, TS), population based methods (evolutionary algorithms; GA, Particle Swarm Optimization (PSO), and scatter search), Ant Colony Optimization (ACO) are the notable ones that are used to solve CFP.

Artificial Immune System (AIS) algorithms imitate the main characteristics of immune systems such as recognition, robustness, feature extraction, diversity, reinforcement learning, memory, and adaptiveness have attracted the attention of researchers (Dasgupta et al. 2003). Basic immune models and algorithms studied in the literature are: Bone Marrow Models, Negative Selection Algorithms, Clonal Selection Algorithm, Somatic Hypermutation, and Immune Network

Models (De Castro and Von Zuben 1999). Brownlee (2011) also summarizes basic steps of the immune based algorithms.

The problem types that are mainly studied by means of CSA are grouped in Ulutas and Kulturel-Konak (2011) as follows: function optimization (i.e., multi modal optimization and continuous function optimization), pattern recognition (i.e., binary character and face detection), design (i.e., continuous design, electromagnetic design, and hardware/software design), scheduling (i.e., job shop scheduling and project scheduling), industrial engineering related problems (i.e., facility location, layout, assembly planning, and material handling systems), TSP, and others (i.e., time series prediction, classification, fault diagnosis, machine learning, and virus detection, etc.). This paper improves the CSA encoding for CFP.

## Cell formation problem

A cellular manufacturing system considers set of part types, part type demand and machines (resources) and basically design problems deal with forming part families according to their processing requirements, grouping machines into manufacturing cells and assigning part families to cells. Columns of a binary machine-part incidence matrix represent machines and rows represent parts. If a machine is used to process a part it is stated as “1” in the matrix and if it is not used, represented as “0”. A perfect grouping that is obtained by grouping all “1”’s in the diagonal blocks on part-machine incidence matrix. The exceptional elements represents the inter-cell movement that corresponds to “1”’s out of the diagonal blocks. “0”’s in the diagonal blocks state that the part does not require related machine type in the cell is named as voids. The utilization of the cell gets lower when the number of voids increase. The performance of a CFP algorithm can be measured by efficiency and efficacy. Efficacy value is calculated by Eq. (1) that is provided by Kumar and Chandrasekharan (1990).

$$\Gamma = \frac{1 - \Psi}{1 + \Phi} \quad (1)$$

where,

$$\Psi = \frac{\text{\# of exceptional elements}}{\text{total \# of operations}} \quad (2)$$

$$\Phi = \frac{\text{\# of voids in the diagonal blocks}}{\text{total \# of operations}} \quad (3)$$

The grouping is better when the value is closer to 1. The grouping efficacy value ranges between 0 and 1. An example with 4 machines and 5 parts is given to illustrate block diagonalization of an incidence matrix. Figure 1a represents the incidence matrix where total number of operations is 9. Figure 1b represents the block diagonalized matrix. Since the

number of voids in the diagonal block is 1 and the number of exceptional elements is 0, efficacy value for the example is calculated as 0.9.

Tunukij and Hicks (2009) state that each cell must contain at least one part and one machine.

A cell design that has less than two machines or parts (i.e., with only one machine and part) is named as singleton cell as represented in Fig. 2a. The cell design has 6 voids and 2 exceptional elements. Therefore, efficacy value is calculated as 0.4665. On the other hand, if a machine or part is not assigned to a cell it is named as residual cell. Figure 2b illustrates the cell design that has 9 voids and 5 exceptional elements. Efficacy value is calculated as 0.2222. Even this small sized example can prove how singleton and residual cell design adversely affect efficacy value.

## Proposed clonal selection algorithm

In this study, a clonal selection based method is defined for CFP that is one of the population based algorithms of AIS. Basic steps of the algorithm in concern is adopted from Talbi (2009) as follows:

**Input:** Initial population  $P_0$ .

$P = P_0 ; /* Generation of the initial population of random antibodies */$

**Repeat**

Evaluate all existing antibodies and compute their affinities;

Select N% of antibodies with highest affinities;

Clone the selected antibodies;

Mature the cloned antibodies (mutation operator);

Evaluate all cloned antibodies;

Add R% of best cloned antibodies to the pool of antibodies;

Remove worst members of the antibodies pool (receptor editing);

Add new random antibodies into the population;

Until stopping criteria satisfied

**Output:** Best population found.

**Encoding** Parts index  $i$  is defined as to take values from 1 to  $p$  and machines index  $j$  takes values from 1 to  $m$ . The number of cells cannot be greater than  $m$  and  $p$ . Cell index  $k$  takes values from 1 to  $c = \min(p, m)$  because every cell should contain at least one machine and one part. Then,  $\text{MaxCell} = \min(\#\text{parts}, \#\text{machines})$ . Cell number is represented as “0” in the encoding that is randomly generated. The number of “0” in an encoding is equal to  $\text{MaxCell}-1$ .

This encoding is represented as [number of parts+number of machines| 0 to identify cells].

**Example** Let us consider a problem with 4 parts and 8 machines. Parts array can be defined as {1, 2, 3, 4} and machines as {1, 2, 3, 4, 5, 6, 7, 8}.  $\text{MaxCellNumber} = \min(4, 8) = 4$  for this example. Therefore, three “0”’s are included to identify the cells ( $\text{MaxCell} - 1 = 4 - 1 = 3$ ). The encoding can be represented as {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 0, 0, 0} where first 4 numbers correspond to part number and the rest defines the machines.

An antibody can randomly be generated as {1, 2, 5, 6, 0, 3, 9, 10, 11, 0, 4, 12, 7, 8, 0} stating that Cell#1 includes parts

**Fig. 1** **a** Part-machine incidence matrix, **b** block-diagonalized matrix

(a)	M1	M2	M3	M4
P1	0	1	0	1
P2	1	0	1	0
P3	0	1	0	1
P4	1	0	1	0
P5	1	0	0	0

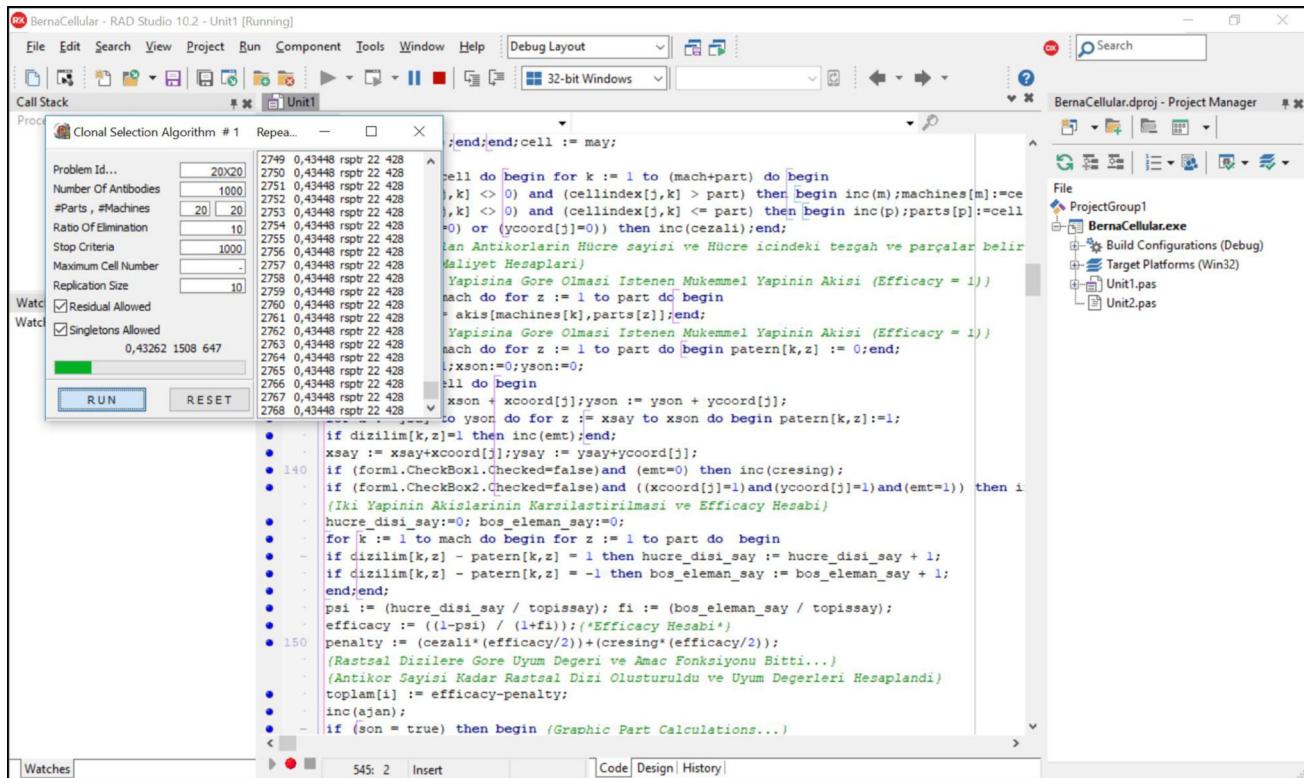
(b)	M1	M3	M2	M4
P2	1	1	0	0
P4	1	1	0	0
P5	1	0	0	0
P1	0	0	1	1
P3	0	0	1	1

**Fig. 2** **a** Representing singleton cell, **b** representing residual cell

(a)	M1	M2	M3	M4
P5	1	0	0	0
P1	0	1	0	1
P2	1	0	1	0
P3	0	1	0	1
P4	1	0	1	0

(b)	M1	M3	M2	M4
P3	0	0	1	1
P2	1	1	0	0
P4	1	1	0	0
P5	1	0	0	0
P1	0	0	1	1



**Fig. 3** A screen shot for the developed CSA when running a test problem

1, 2 and machines 1, 2; Cell#2 includes part 3, machines 5, 6, 7; Cell#3 includes part 4 and machines 3, 4, 8. Since each part and machine is assigned to a cell there is no need for a repair function.

The encoding enables to form cells without the prior knowledge of results in the literature. Main contribution and superiority of the representation and encoding is its flexibility.

**Objective** The aim in CFP is to group parts and machines so that all parts in a family are processed within a machine group with minimum interaction with other groups. In this paper, grouping efficacy is chosen as the measure of perfor-

mance since several studies in the literature provide efficacy results as a standard measure to report the quality of the grouping.

**Selection Roulette Wheel** selection principle that is introduced by Goldberg (1989) is used. All individuals in the population have a chance to be selected. However, antibodies that have higher efficacy values have a higher selection probability.

**Mutation operator** A pairwise interchange mutation is utilized to matured selected antibodies. First, two locations on an antibody are selected randomly. Then, the values at these locations are interchanged. The objective function is calcu-

**Table 1** Data set A for cell formation problem

Problem no	Data set source	Machine	Part
A1	King and Nakornchai (1982)	5	7
A2	Waghodekar and Sahu (1984)	5	7
A3	Seifoddini (1989)	5	18
A4	Kusiak and Cho (1992)	6	8
A5	Kusiak and Chow (1987)	7	11
A6	Boctor (1991)	7	11
A7	Seifoddini and Wolfe (1986)	8	12
A8	Chandrasekaran and Rajagopalan (1986a)	8	20
A9	Chandrasekaran and Rajagopalan (1986b)	8	20
A10	Mosier and Taube (1985a)	10	10
A11	Chan and Milner (1982)	15	10
A12	Askin and Subramanian (1987)	14	24
A13	Stanfel (1985)	14	24
A14	McCormick et al. (1972)	16	24
A15	Srinivasan et al. (1990)	16	30
A16	King (1980)	16	43
A17	Carrie (1973)	18	24
A18	Mosier and Taube (1985b)	20	20
A19	Kumar et al. (1986)	23	20
A20	Carrie (1973)	20	35
A21	Boe and Cheng (1991)	20	35
A22	Chandrasekharan and Rajagopalan (1989)—Dataset 1	24	40
A23	Chandrasekharan and Rajagopalan (1989)—Dataset 2	24	40
A24	Chandrasekharan and Rajagopalan (1989)—Dataset 3	24	40
A25	Chandrasekharan and Rajagopalan (1989)—Dataset 5	24	40
A26	Chandrasekharan and Rajagopalan (1989)—Dataset 6	24	40
A27	Chandrasekharan and Rajagopalan (1989)—Dataset 7	24	40
A28	McCormick et al. (1972)	27	27
A29	Carrie (1973)	28	46
A30	Kumar and Vannelli (1987)	30	41
A31	Stanfel (1985)—Figure 5	30	50
A32	Stanfel (1985)—Figure 6	30	50
A33	King and Nakornchai (1982)	30	90
A34	McCormick et al. (1972)	37	53
A35	Chandrasekharan and Rajagopalan (1987)	40	100

lated for the new encoding and if new efficacy value is higher, antibody is updated.

**Example 1** if the locations 3 and 9 are selected {1, 2, **5**, 6, **0**, 3, 9, 10, **11**, **0**, 4, 12, 7, 8, **0**} then the antibody changes as {1, 2, **11**, 6, **0**, 3, 9, 10, **5**, **0**, 4, 12, 7, 8, **0**} stating that Cell#1 includes parts 1, 2 and machines 7, 2; Cell#2 includes part 3, machines 5, 6, 9; Cell#3 includes part 4 and machines 3, 4, 8.

**Example 2** if the locations 5 and 8 are selected {1, 2, 5, 6, **0**, 3, 9, **10**, 11, **0**, 4, 12, 7, 8, **0**} the antibody changes as {1, 2, 6, **10**, 3, 9, **0**, 11, **0**, 4, 12, 7, 8, **0**} stating that Cell#1 includes parts 1,

2, 3 and machines 2, 5, 6; Cell#2 includes machine 7; Cell#3 includes part 4, machines 8, 3, and 4. During calculation of the efficacy value, a penalty is defined because Cell#2 does not include any part. Likewise, if a cell does not include any machine, a penalty value is defined and antibody's objective function value gets a lower efficacy value. The penalty value reduces the chance of the antibody to be selected in the next iteration.

**Example 3** since the locations are selected randomly, the antibody {1, 2, 5, 6, **0**, 3, 9, 10, 11, **0**, 4, 12, 7, **8**, **0**} may change as {1, 2, 5, 6, **0**, 3, 9, 10, 11, **8**, 4, 12, 7, 8, **0**, **0**} when the locations of 10 and 14 are selected. This antibody now rep-

**Table 2** CFP results for Data set A

No	Best known so far	CSA # of cells	CSA efficacy value			CSA # of iterations	
			Best	Avg.	Worst	Best	Total
A1	0.8235	2	0.8235	0.8235	0.8235	1.20	1002.20
A2	0.6957	2	0.6957	0.6957	0.6957	0.80	1001.80
A3	0.7959	2	0.7959	0.7959	0.7959	8.20	1009.20
A4	0.7692	2	0.7692	0.7692	0.7692	1.60	1002.60
A5	0.6087	4	0.6087	0.6087	0.6087	23.90	1024.90
A6	0.7083	5	0.7083	0.7083	0.7083	14.60	1015.60
A7	0.6944	4	0.6944	0.6944	0.6944	27.20	1028.20
A8	0.8525	3	0.8525	0.8525	0.8525	21.50	1022.50
A9	0.5872	2	0.5872	0.5872	0.5872	33.50	1034.50
A10	0.7500	5	0.7500	0.7500	0.7500	14.90	1015.90
A11	0.9200	3	0.9200	0.9200	0.9200	15.20	1016.20
A12	0.7206	7	0.7206	0.7206	0.7206	124.50	1125.50
A13	0.7183	7	0.7183	0.7183	0.7183	111.10	1112.10
A14	0.5326	8	0.5326	0.5326	0.5326	665.40	1666.40
A15	0.6899	6	0.6899	0.6899	0.6899	187.60	1188.60
A16	0.5753	8	0.5753	0.5745	0.5732	1256.80	2360.90
A17	0.5773	9	0.5773	0.5761	0.5728	875.80	1876.80
A18	0.4345	5	0.4345	0.4343	0.4326	1134.40	2135.40
A19	0.5081	7	0.5081	0.5021	0.4965	1205.00	2206.00
A20	0.7791	5	0.7791	0.7791	0.7791	582.00	1583.00
A21	0.5798	5	0.5798	0.5798	0.5798	750.40	1751.40
A22	1	7	1	1	1	379.80	1380.80
A23	0.8511	7	0.8511	0.8511	0.8511	503.90	1504.90
A24	0.7351	7	0.7351	0.7351	0.7351	651.90	1652.90
A25	0.5329	11	0.5329	0.5323	0.5290	1310.30	2311.30
A26	0.4895	12	0.4895	0.4867	0.4832	5338.71	7476.00
A27	n/a	11	0.4631	0.4600	0.4533	3018.80	4019.80
A28	0.5482	5	0.5482	0.5482	0.5482	429.70	1430.70
A29	n/a	11	0.4691	0.4655	0.4624	2805.80	3806.80
A30	0.6331	13	0.6331	0.6264	0.6242	2163.50	3164.50
A31	0.5977	13	0.5977	0.5967	0.5917	2240.90	3241.90
A32	n/a	14	0.5083	0.5062	0.5026	2898.60	3899.60
A33	0.4800	16	0.4801	0.4782	0.4755	3365.80	4070.20
A34	0.6064	3	0.6064	0.6014	0.6008	1473.20	2474.20
A35	0.8403	10	0.8403	0.8403	0.8403	1875.80	2876.80

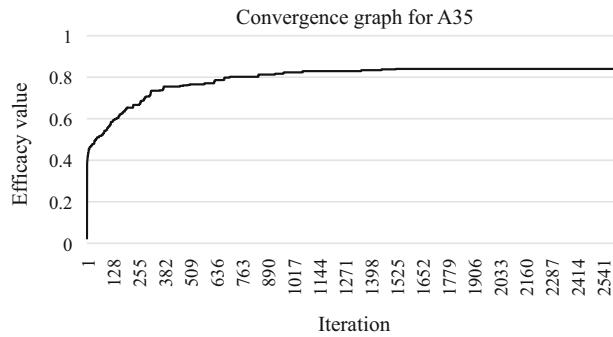
resents two cells where Cell#1 includes parts 1, 2, machines 1, 2 and Cell#2 includes part 3, 4, machines, 3, 4, 5, 6, 7, 8.

**Receptor editing operator** After mutation processes, the antibodies that have worse efficacy values are erased (worst %B of the whole population). Then, same percent of new antibodies are randomly generated. By this means, it is possible to search new regions in the total search space.

**Termination criterion** The algorithm is terminated when there is no improvement after a certain number (i.e., 1000) of iterations.

## Computational results

CSA is coded in licensed Embarcadero Delphi 10.2 and run on an Intel Core i3 machine with 1.60 GHz CPU and 4.00 Gb of memory. The parameters for CSA are defined as population size, receptor editing, and termination criteria when solving both data sets (Data set A and Data set B). Related values are determined based on prior computational experiments. Problem size is considered as the main criteria when assigning values for population size and termination criteria.

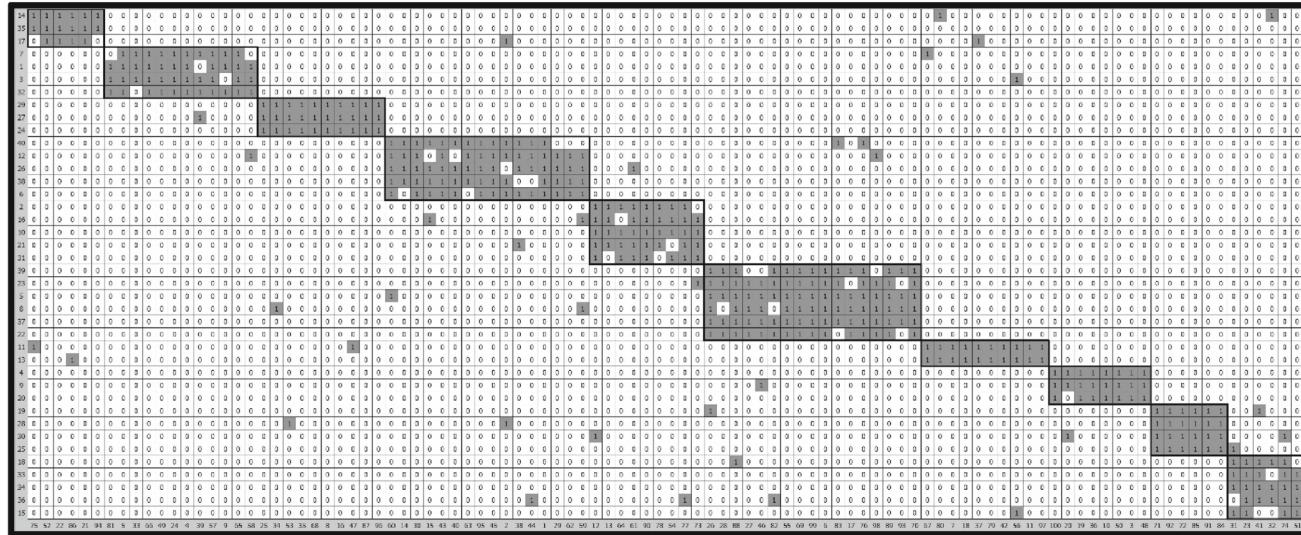


**Fig. 4** Convergence graph for A25

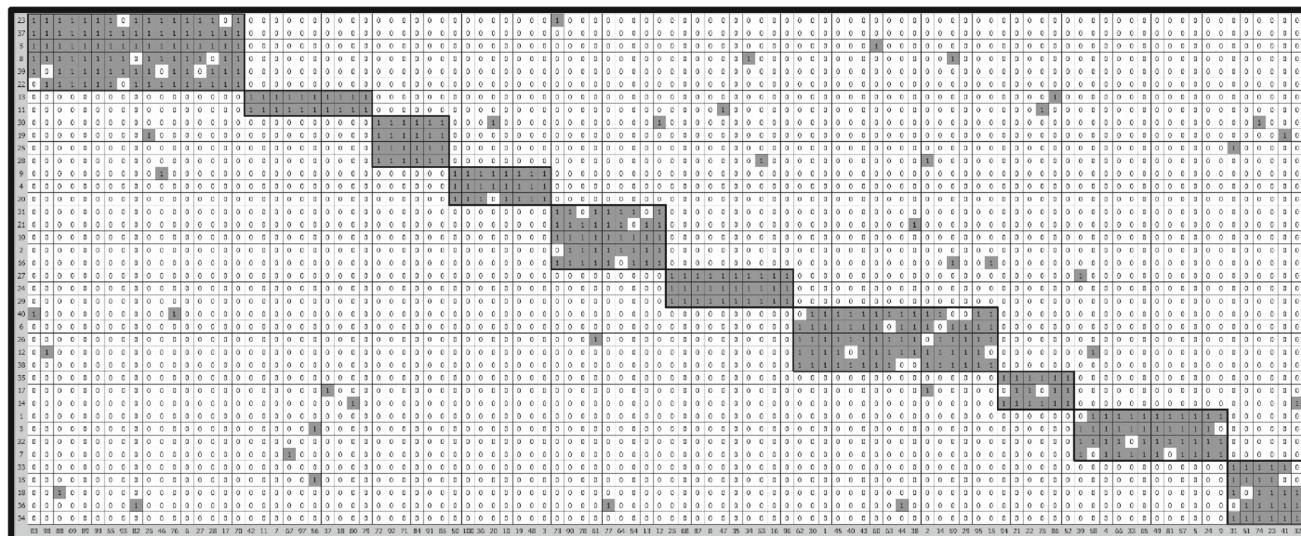
Population size is assigned as 100 for small sized (A1–A15 and B1–B25) and as 1000 for large sized (A16–A35 and B26–B32) test problems. It is known that small population size may adversely influence the quality of the solution for it is related with the search space. On the other hand, when large population size is defined, computation time may increase.

Defining a certain number of iterations may lead to poor solutions especially for large sized problems. Algorithm is terminated if no improvement is obtained in the efficacy value after 1000 iterations. Receptor editing operator enables to eliminate worst antibodies and diversify the solution with the new generated antibodies. Ulutas and Islier (2007) discuss

**(a)**



**(b)**



**Fig. 5** **a** Alternative CSA grouping for A35 Efficacy value: 0.8403, number of cells: 10, **b** alternative CSA grouping for A35 Efficacy value: 0.8403, number of cells: 10

that defining large value (i.e., 50, 30%) for receptor editing parameter may cause losing the solutions that have a potential to mature. On the other hand, small value (i.e., 1, 5%) may not help to diversify the population. Therefore, receptor editing percent is defined as 10% for all test problems in concern.

Figure 3 represents a screen shot for the developed CSA. The test problem data (part-machine incidence matrix) is retrieved from a file based on the defined “Problem id”. Related values are assigned for CSA parameters; “Number of antibodies”, “Ratio of elimination”, and “Stop criteria”. Major restriction is the definition of a fixed number of production cells in the literature solutions. It is possible to define a “Maximum cell number” for the developed algorithm. However, solutions are generated without the definition of a fixed cell number. Exactly same efficacy values and cell numbers are obtained for small sized test problems. On the other hand, the flexibility of the encoding enabled the algorithm to successfully generate alternative cell designs with different cell numbers, especially for large sized test problems. “Replication size” aids a researcher to define how many times the algorithm will run. The restrictions for residual and singleton cells can be defined easily by use of the check boxes in the interactive data entry screen. It is also possible to track the progress of the solution.

A text file is created to represent the basic block-diagonalized matrix, best efficacy value, and CPU time for each run. Efficacy values calculated in each iteration are saved in another text file. Likewise, final cell grouping and part-machine assignment for each run is illustrated in an image file. Final CSA results are summarized in a separate text file including the values of replication number, number of cells, best efficacy value, iteration number (best value obtained), Total iteration number, CPU time (best value obtained), and Total CPU time.

## Computational results for data set A

The data set provided by Goncalves and Resende (2004) is corrected and named as Data set A by Bychkov and Batsyn (2018). Table 1 summarizes the well-known cell formation dataset that includes 35 test instances with ranging size of 5 to 40 machines and 7 to 100 parts. Sources of the test problems and related number of machines and parts are also given in relevant columns.

Bychkov et al. (2013) provide a heuristic algorithm and Bychkov et al. (2014) focus on exact model for the CFP. CSA results are compared with the most recent CFP research in the literature where mixed integer linear programming model is used (Bychkov and Batsyn 2018). It is stated that the model that does not include a constraint for residual cells. The drawback of their study is not providing any information about the

**Table 3** Data set B for cell formation problem

Problem no	Data set source	Machine	Part
B1	Adil et al. (1996)	6	6
B2	Pa Rkin and Li (1997)	6	7
B3	Brown and Sumichrast (2001)	6	11
B4	Chan and Milner (1982)	7	5
B5	Kusiak and Chow (1987)	7	8
B6	Zolfaghari and Liang (2002)	7	8
B7	Won and Kim (1997)	7	10
B8	Sarker and Khan (2001)	8	8
B9	Nair (1999)	8	10
B10	Islam and Sarker (2000)	8	10
B11	Kumar et al. (1986)	9	15
B12	Ham et al. (1985)	10	8
B13	Viswanathan (1996)	10	12
B14	Shargal et al. (1995)	10	38
B15	Won and Kim (1997)	11	10
B16	Seifoddini (1989)	11	22
B17	Moon and Chi (1992)	12	19
B18	Li (2003)	14	14
B19	Chan and Milner (1982)	15	10
B20	Yang and Yang (2008)	15	15
B21	Yang and Yang (2008)	15	15
B22	Yang and Yang (2008)	15	15
B23	Harhalakis et al. (1994)	17	20
B24	Seifoddini and Djassemi (1991)	18	24
B25	Sandbothe (1998)	20	10
B26	Nagi et al. (1990)	20	51
B27	Won and Kim (1997)	26	28
B28	Yang and Yang (2008)	28	35
B29	Seifoddini and Djassemi (1996)	35	15
B30	Seifoddini and Djassemi (1996)	41	50
B31	Yang and Yang (2008)	46	105
B32	Zolfaghari and Liang (1997)	50	150

number of cells in the final cell design. Test problem data are available upon request for the interested researchers.

The number of cells generated for the best CSA efficacy value are given in the third column of Table 2. Efficacy values obtained from 10 replications for each test problem are illustrated as maximum/best, average, and minimum/worst in the following columns. The iteration when the best solution is obtained and the total number of iterations (based on the termination criteria) are also recorded during the experiments for each test problem. Average of 10 replications are stated as the best and total iterations in the table. CSA is able to reach the best efficacy values for the 32 test problems among Data set A. It is stated that CPLEX runs too long or takes too much memory, therefore, Bychkov and Batsyn (2018) have not provided results for test problems A27, A29 and A32.

**Table 4** CFP results for Data set B

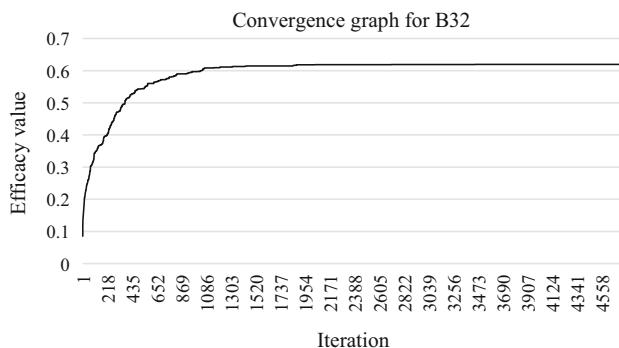
No	Best known so far	CSA # of cells	CSA efficacy value			CSA # of iterations	
			Best	Avg.	Worst	Best found	Total
B1	0.8095	2	0.8095	0.8095	0.8095	3.70	1004.70
B2	0.7222	3	0.7222	0.7222	0.7222	14.80	1015.80
B3	0.6071	2	0.6071	0.6071	0.6071	30.80	1031.80
B4	0.8889	4	0.8889	0.8889	0.8889	3.20	1004.20
B5	0.7500	3	0.7500	0.7500	0.7500	13.90	1014.90
B6	0.7391	3	0.7391	0.7391	0.7391	10.80	1011.80
B7	0.8148	5	0.8148	0.8148	0.8148	15.40	1016.40
B8	0.7222	3	0.7222	0.7222	0.7222	41.90	1042.90
B9	0.7576	5	0.7576	0.7576	0.7576	19.70	1020.70
B10	0.9000	3	0.9000	0.9000	0.9000	14.50	1015.50
B11	0.7273	6	0.7273	0.7273	0.7273	288.70	1289.70
B12	0.8276	3	0.8276	0.8276	0.8276	15.40	1016.40
B13	0.5962	3	0.5962	0.5962	0.5962	84.10	1085.10
B14	<b>0.6405</b>	6	0.6404	0.6404	0.6404	330.50	1331.50
B15	0.8333	4	0.8333	0.8333	0.8333	27.10	1028.10
B16	0.7391	3	0.7391	0.7391	0.7391	76.90	1077.90
B17	0.6552	6	0.6552	0.6552	0.6552	220.70	1221.70
B18	0.6129	4	0.6129	0.6129	0.6129	226.30	1227.30
B19	0.8000	3	0.8000	0.8000	0.8000	36.10	1037.10
B20	0.8710	4	0.8710	0.8710	0.8710	69.40	1070.40
B21	0.8333	4	0.8333	0.8333	0.8333	82.30	1083.30
B22	0.7258	4	0.7258	0.7258	0.7258	96.60	1097.60
B23	0.8111	4	0.8111	0.8111	0.8111	133.00	1134.00
B24	0.5673	8	0.5673	0.5673	0.5673	753.60	1754.60
B25	0.7600	4	0.7600	0.7600	0.7600	74.50	1075.50
B26	0.6068	8	0.6068	0.60591	0.6034	1375.50	2376.50
B27	0.7248	6	0.7248	0.7248	0.7248	320.30	1321.30
B28	0.6729	6	0.6729	0.6729	0.6729	465.80	1466.80
B29	0.5730	10	0.5730	0.57239	0.5700	1324.60	2325.60
B30	0.7308	18	0.7308	0.72328	0.7143	2047.20	2548.20
B31	0.6798	8	0.6798	0.67668	0.6701	1815.80	2066.80
B32	n/a	7	<b>0.6193</b>	0.61902	0.6183	4611.00	5612.00

The values written in bold represent best values

Since, the hardware platform and solution approach are not same, no comparison for CPU time is given. The grouping figures for Data set A are provided in “Appendix I” to enable the researchers make comparisons.

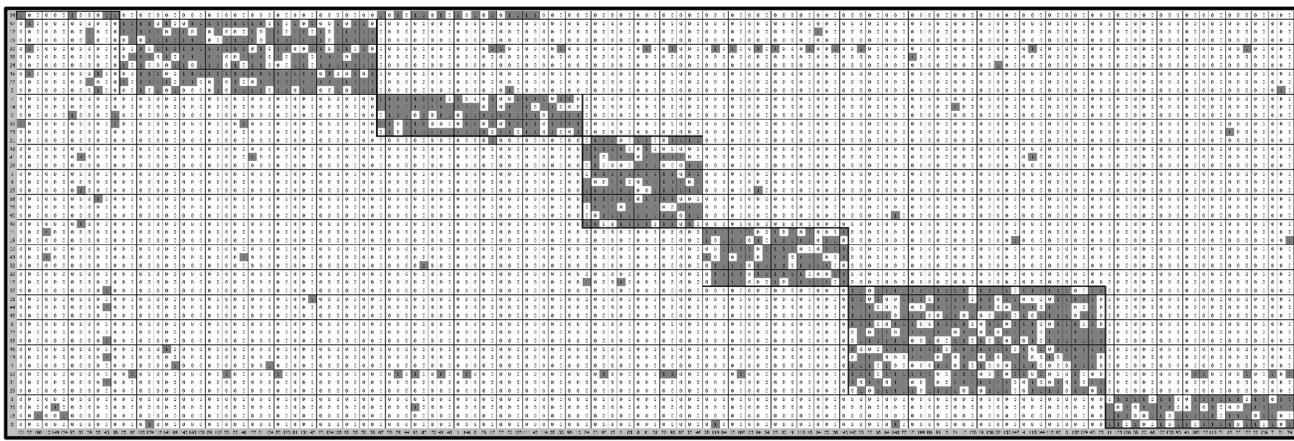
The best efficacy value for A35 (the largest test problem in data set A) is obtained as 0.8403. The convergence graph for A35 is illustrated in Fig. 4 where the best value is obtained in the 1523th iteration and the algorithm is terminated in the 2662th iteration. The average of total CPU time is 2114.1 for A35 where it is 58.1 for A1 (the smallest test problem in the data test).

Figure 5a, b represent alternative grouping examples for A25 where best value is obtained with 10 cells by CSA. Final grouping does not include any residual or singleton cell.

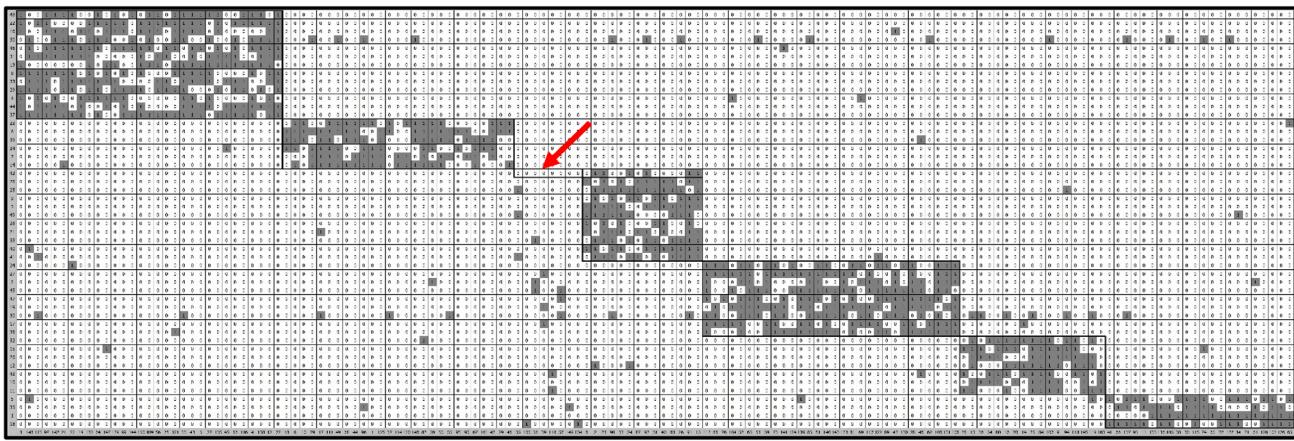


**Fig. 6** Convergence graph for B32 best efficacy value

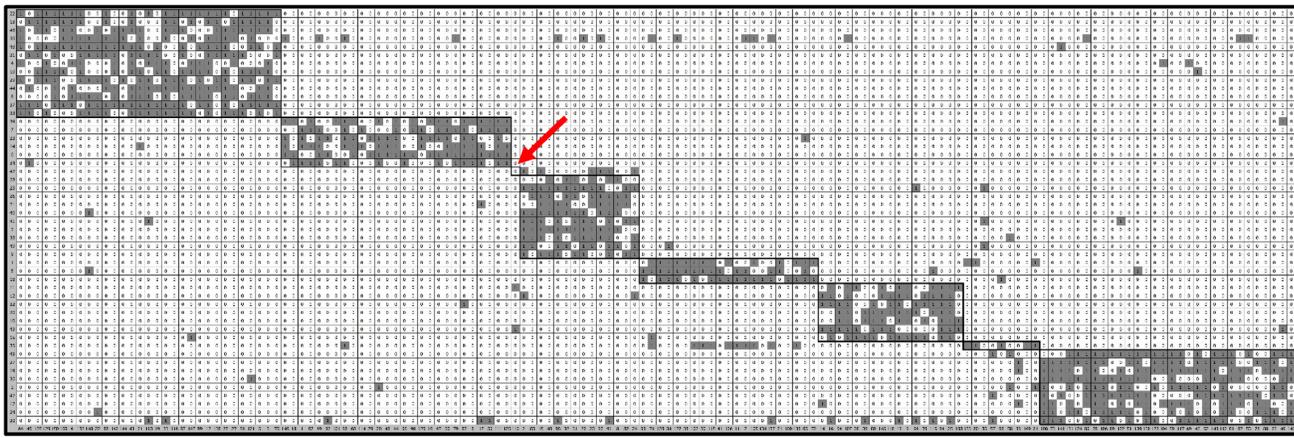
(a)



(b)



(c)



**Fig. 7** **a** B32 result: Efficacy value: 0.6193, number of cells: 7, number of residual cells: 0, **b** B32 result: Efficacy value: 0.6191, number of cells: 7, number of residual cells: 1, **c** B32 result: Efficacy value: 0.6183, number of cells: 8, number of residual cells: 1

## Computational results for data set B

Data set B are less popular in research papers and consists of 32 test instances with machine size ranging from 6 to 50 and part size ranging from 6 to 150 (Table 3). Data considered in this study are accessed from Bychkov and Batsyn (2018) and available for the interested researchers upon request.

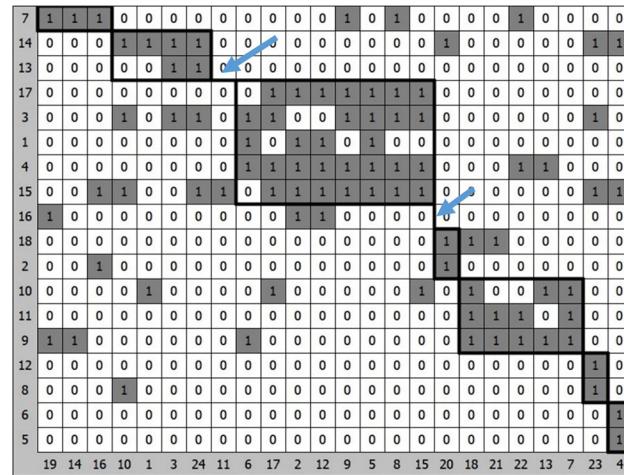
Table 4 provides best known results from Bychkov and Batsyn (2018) and maximum/best, average, and minimum/worst results obtained by CSA and the number of cells. CSA is able to reach the best efficacy values for the 30 test problems among Data set B. Only, efficacy value for B14 is slightly higher ( $= 0.0156\%$ ) than CSA value that be resulting from a rounding. All grouping figures obtained by CSA for data set B are provided in “Appendix II” to enable the researchers make comparisons.

Bychkov and Batsyn (2018) state that due to the high computational complexity, it was not possible to solve the largest test problem (B32,  $50 \times 150$ ). However, CSA generated solutions for B32 in rather short CPU time ( $= 5173$ ). The convergence graph for B32 best efficacy value ( $= 0.6193$ ) is obtained in the 3449th iteration (Fig. 6). Since the termination criteria is defined as “terminate after no improvements are obtained after 1000 iterations”, the algorithm is terminated in the 4773th iteration.

To assess the cell formation grouping, it is important to illustrate the solution by graphs. Especially when solving a real size-sized problem, a grouping without a machines or parts (residual cell) and a cell with only one machine and part (singleton cell) can be problematic. Figure 7a, b illustrate the graphical analysis of CSA B32 results based on number of cells and residual cells. Residual cells in Fig. 7b, c are pointed with an arrow.

B11, B13, B14, B16, B17, B19, B24, B25, B26, B29, B30 results of Bychkov and Batsyn (2018) solved with the model that allow residual cells are reported generate better results (higher efficacy values) compared to the model that does not allow residual cells. However, it is identified that wrong definition of the residual cell may lead to ill-structured grouping. A cell (either named as singleton or residual) can be formed with the assignment of at least one part and machine.

CSA is run after removing related constraint that force assigning at least a machine and a part to a cell. Figure 8 represents a solution obtained for B24 by CSA as an example. Please note that Part 11 is not associated with a machine and



**Fig. 8** B24 result: Efficacy value: 0.5728, number of cells: 9, number of residual cells: 0

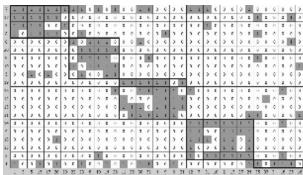
Machine 16 is not associated with a part. Therefore, results of Bychkov and Batsyn (2018) with the model that is claimed to allow residual cells are disregarded in this study.

## Conclusions

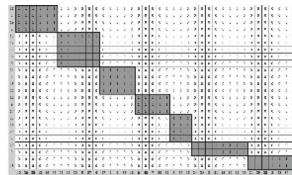
Today’s manufacturing environment requires to deal with big data and provide quick and practical solutions. Several methodologies are introduced to solve CFP in the literature. Although mathematical models can generate exact solutions, due to large number of variables and constraints, long computational times may be required to solve even small-sized CFP instances in platforms such as CPLEX, GAMS, Lingo etc. This study introduces a CSA with a novel encoding. Results illustrate that the algorithm is able to obtain best efficacy values for the 67 test problems in concern and it is powerful for it can deal with large-sized problems. Visual representation of the CSA results enable to validate and verify the results. Solution quality and definition for residual cell is criticized along with an example for the solutions that are claimed to generate better results with the mixed-integer linear programming model that allow residual cells. Future studies may focus on solving CFP problems with data from real life problems. Because the basic interactive software, currently developed to solve test problems, has a potential to be used for industrial applications.



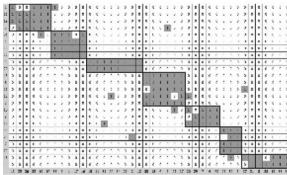
A21



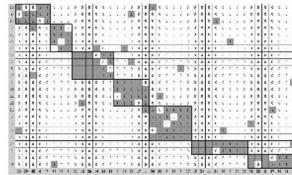
A22



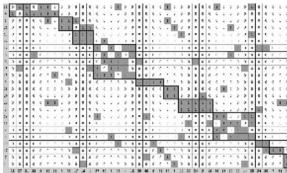
A23



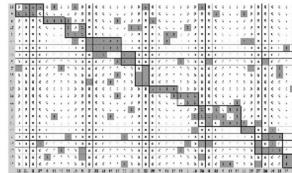
A24



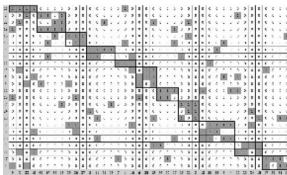
A25



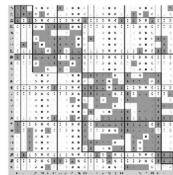
A26



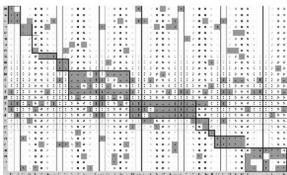
A27



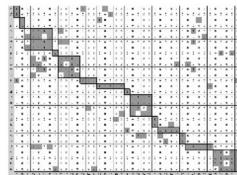
A28



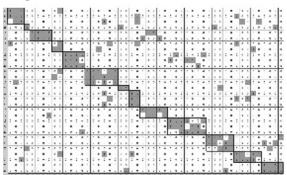
A29



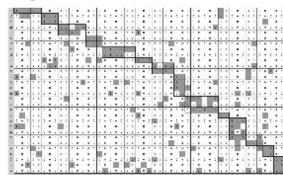
A30



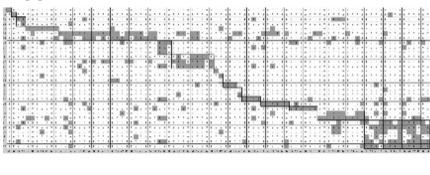
A31



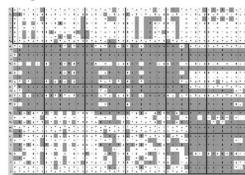
A32



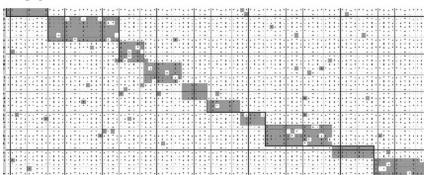
A33



A34

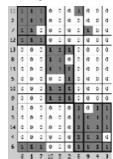


A35

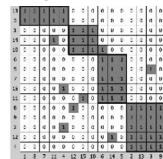




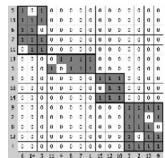
B19



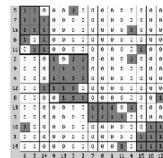
B20



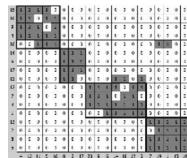
B21



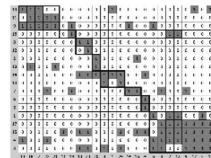
B22



B23



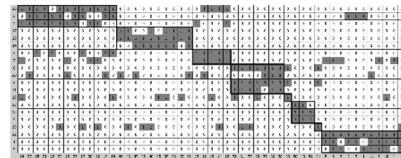
B24



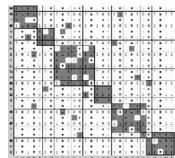
B25



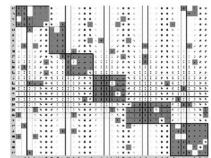
B26



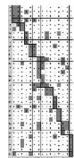
B27



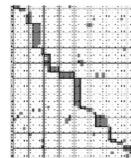
B28



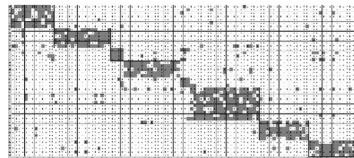
B29



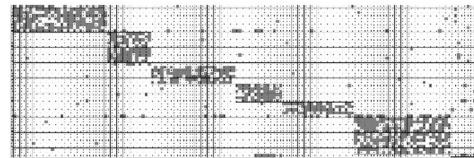
B30



B31



B32



## References

- Adil, G. K., Rajamani, D., & Strong, D. (1996). Cell formation considering alternate routings. *International Journal of Production Research*, 34(5), 1361–1380.
- Arkat, A., Hosseini, L., & Hosseinabadi, F. M. (2011). Minimization of exceptional elements and voids in the cell formation problem using a multi-objective genetic algorithm. *Expert Systems with Applications*, 38, 9597–9602.
- Arvindh, B., & Irani, S. A. (1994). Cell formation: The need for an integrated solution of the sub problem. *International Journal of Production Research*, 32(5), 1197–1218.
- Askin, R. G., & Subramanian, S. (1987). A cost-based heuristic for group technology configuration. *International Journal of Production Research*, 25, 101–113.
- Boctor, F. A. (1991). Linear formulation of the machine-part cell formation problem. *International Journal of Production Research*, 29(2), 343–356.
- Boe, W., & Cheng, C. H. (1991). A close neighbor algorithm for designing cellular manufacturing systems. *International Journal of Production Research*, 29(10), 2097–2116.
- Brownlee J. (2011). Clever algorithms: Nature-inspired programming recipes. *Creative Commons* 280, <http://www.cleveralgorithms.com/>. Last accessed February 2018.
- Brown, E., & Sumichrast, R. (2001). CF-GGA: a grouping genetic algorithm for the cell formation problem. *International Journal of Production Research*, 36, 3651–3669.
- Bychkov, I., & Batsyn, M. (2018). An efficient exact model for the cell formation problem with a variable number of production cells. *Computers and Operations Research*, 91, 112–120.
- Bychkov, I. S., Batsyn, M. V., & Pardalos, P. M. (2014). Exact model for the cell formation problem. *Optimization Letters*, 8(8), 2203–2210.
- Bychkov, I., Batsyn, M., Sukhov, P., & Pardalos, P. M. (2013). Heuristic algorithm for the cell formation problem. Models, algorithms, and technologies for network analysis. *Springer Proceedings in Mathematics & Statistics*, 59, 43–69.
- Carrie, S. (1973). Numerical taxonomy applied to group technology & plant layout. *International Journal of Production Research*, 11, 399–416.
- Chandrasekaran, M. P., & Rajagopalan, R. (1986a). An ideal seed nonhierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24, 451–463.
- Chandrasekaran, M. P., & Rajagopalan, R. (1986b). MODROC: An extension of rank order clustering of group technology. *International Journal of Production Research*, 24(5), 1221–1233.
- Chandrasekharan, M. P., & Rajagopalan, R. (1987). ZODIAC: An algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research*, 24(2), 835–850.
- Chandrasekharan, M. P., & Rajagopalan, R. (1989). Groupability: Analysis of the properties of binary data matrices for group technology. *International Journal of Production Research*, 27(6), 1035–1052.
- Chan, H. M., & Milner, D. A. (1982). Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing System*, 1, 65–75.
- Cheng, C., Gupta, Y., Lee, W., & Wong, K. (1998). A TSP-based heuristic for forming machine groups and part families. *International Journal of Production Research*, 36, 1325–1337.
- Dasgupta, D., Ji, Z., & Gonzalez, F. (2003). Artificial immune systems research in the last five years. In *Proceedings of the 2003 congress on evolutionary computation conference*. Canberra. Australia. (pp. 123–130). December 8–13.
- De Castro, L., & Von Zuben, F. J. (1999). Artificial immune systems: Part I: Basic theory & applications. FEEC Univ. Campinas, Campinas, Brasil, [ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/tr\\_dca/trdca0199.pdf](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/tr_dca/trdca0199.pdf). Last accessed January 2018.
- Dimopoulos, C., & Mort, N. (2001). A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. *International Journal of Production Research*, 39, 1–19.
- Dimopoulos, C., & Zalzala, A. (2000). Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. *IEEE Transactions on Evolutionary Computation*, 4(2), 93–113.
- Farahani, M. H., & Hosseini, L. (2011). An ant colony optimization for the machine-part cell formation problem. *International Journal of Computational Intelligence Systems*, 4, 486–496.
- Goldberg, D. E. (1989). *Genetic algorithms: Search, optimization and machine learning*. Boston, MA: Addison-Wesley Longman Publishing Co. Inc.
- Goncalves, J. F., & Resende, M. G. C. (2004). An evolutionary algorithm for manufacturing cell formation. *Computers and Industrial Engineering*, 47, 247–273.
- Ham I., Hitomi K., & Yoshida T. (1985). Layout planning for group technology. In *Group technology. International series in management science/operations research*. (pp. 153–169). Dordrecht: Springer.
- Harhalakis, G., Ioannou, G., Minis, I., & Nagi, R. (1994). Manufacturing cell formation under random product demand. *International Journal of Production Research*, 32(1), 47–64.
- Hyer, N., & Wemmerlov, U. (1984). Group technology and productivity. *Harvard Business Review*, 62(4), 140–149.
- Islam, K. M. S., & Sarker, B. R. (2000). A similarity coefficient measure and machine parts grouping in cellular manufacturing systems. *International Journal of Production Research*, 38(3), 699–720.
- Islier, A. A. (2001). Forming manufacturing cells by using genetic algorithm. *Anadolu University Journal of Science and Technology*, 2, 137–157.
- Islier, A. A. (2005). Group technology by ants. *International Journal of Production Research*, 43(5), 913–932.
- James, T., Brown, E. C., & Keeling, K. B. (2007). A hybrid grouping genetic algorithm for the cell formation problem. *Computers and Operations Research*, 34, 2059–2079.
- Joines, J., Culberth, C. T., & King, R. E. (1996). Manufacturing cell design: an integer programming model employing genetic algorithms. *IEE Transactions*, 28, 69–85.
- Kamel, M., Ghenniwa, H., & Liu, T. (1994). Machine assignment and part-families formation using group technology. *Journal of Intelligent Manufacturing*, 5, 225–234.
- King, J. R. (1980). Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. *International Journal of Production Research*, 18(2), 213–232.
- King, J. R., & Nakornchai, V. (1982). Machine-component group formation in group technology: Review and extension. *International Journal of Production Research*, 20(2), 117–133.
- Kumar, C. S., & Chandrasekharan, M. P. (1990). Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research*, 28, 233–243.
- Kumar, K. R., Kusiak, A., & Vannelli, A. (1986). Grouping of parts and components in flexible manufacturing systems. *European Journal of Operations Research*, 24, 387–397.
- Kumar, K. R., & Vannelli, A. (1987). Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research*, 25(12), 1715–1728.
- Kusiak, A., & Cho, M. (1992). Similarity coefficient algorithm for solving the group technology problem. *International Journal of Production Research*, 30, 2633–2646.
- Kusiak, A., & Chow, W. (1987). Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, 6(2), 117–124.

- Lee, H., Malavi, C. O., & Ramachandran, S. (1992). A self-organizing neural network approach for the design of cellular manufacturing systems. *Journal of Intelligent Manufacturing*, 3, 325–332.
- Li, M. L. (2003). The algorithm for integrating all incidence matrices in multidimensional group technology. *International Journal Production Economics*, 86, 121–131.
- Mahdavi, I., Paydar, M. M., Solimanpur, M., & Heidarzade, A. (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems with Applications*, 36, 6598–6604.
- Malavi, C. O., & Ramachandran, S. (1991). Neural network-based design of cellular manufacturing systems. *Journal of Intelligent Manufacturing*, 2, 305–314.
- McCormick, W. T., Schweitzer, P. J., & White, T. W. (1972). Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20, 993–1009.
- Moon, Y. B. (1992). Establishment of a neurocomputing model for part family/machine group identification. *Journal of Intelligent Manufacturing*, 3, 173–182.
- Moon, Y. B., & Chi, S. C. (1992). Generalized part family formation using neural network techniques. *Journal of Manufacturing Systems*, 11(3), 149–159.
- Mosier, C. T., & Taube, L. (1985a). The facets of group technology & their impact on implementation. *Omega*, 13(6), 381–391.
- Mosier, C. T., & Taube, L. (1985b). Weighted similarity measure heuristics for the group technology machine clustering problem. *Omega*, 13(6), 577–583.
- Nagi, R., Harhalakis, G., & Proth, J. M. (1990). Multiple routeings and capacity considerations in group technology applications. *International Journal of Production Research*, 28(12), 1243–1257.
- Nair, G. J. (1999). Accord: A bicriterion algorithm for cell formation using ordinal and ratio level data. *International Journal of Production Research*, 37(3), 539–556.
- Onwubalu, G. C. (1999). Design of parts for cellular manufacturing using neural network-based approach. *Journal of Intelligent Manufacturing*, 10, 251–265.
- Onwubalu, G. C., & Mutungi, M. (2001). A genetic algorithm approach to cellular manufacturing systems. *Computers and Industrial Engineering*, 39, 125–44.
- Pa Rkin, R. E., & Li, M. L. (1997). The multi-dimensional aspects of a group technology algorithm. *International Journal of Production Research*, 35(8), 2345–2358.
- Papaionnaou, G., & Wilson, J. M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European Journal of Operational Research*, 206, 509–521.
- Sandbothe, R. A. (1998). Two observations on the grouping efficacy measure for goodness of block diagonal forms. *International Journal of Production Research*, 36(11), 3217–3222.
- Sarker, B. R., & Khan, M. (2001). A comparison of existing grouping efficiency measures and a new weighted grouping efficiency measure. *IIE Transactions*, 33, 11–27.
- Seifoddini, H. (1989). Single linkage versus average linkage clustering in machine cells formation applications. *Computers and Industrial Engineering*, 16(3), 419–426.
- Seifoddini, H., & Djassemi, M. (1991). The production data based similarity coefficient versus Jaccard's similarity coefficient. *Computers Industrial Engineering*, 21, 263–266.
- Seifoddini, H., & Djassemi, M. (1996). The threshold value of a quality index for formation of cellular manufacturing systems. *International Journal of Production Research*, 34(12), 3401–3416.
- Seifoddini, H., & Wolfe, P. M. (1986). Application of the similarity coefficient method in group technology. *IIE Transactions*, 18(3), 266–270.
- Shargal, M., Shekhar, S., & Irani, S. A. (1995). Evaluation of search algorithms and clustering efficiency measures for machine-part matrix clustering. *IIE Transactions*, 27(1), 43–59.
- Srinivasan, G. (1994). A clustering algorithm for machine cell formation in group technology using minimum spanning trees. *International Journal of Production Research*, 32(9), 2149–2158.
- Srinivasan, G., Narendran, T., & Mahadevan, B. (1990). An assignment model for the part-families problem in group technology. *International Journal of Production Research*, 28(1), 145–152.
- Stanfel, L. (1985). Machine clustering for economic production. *Engineering Costs and Production Economics*, 9, 73–78.
- Talbi, E. (2009). *Metaheuristics: From design to implementation* (p. 267). New Jersey: Wiley.
- Tunukij, T., & Hicks, C. (2009). An enhanced grouping genetic algorithm for solving the cell formation problem. *International Journal of Production Research*, 47(7), 1989–2007.
- Ulutas, B., & Sarac, T. (2009). A clonal selection algorithm for cell formation problem with alternative routings. In 4<sup>th</sup> international conference of group technology/cellular manufacturing 2009 (GT/CM 2009), 16–18 February 2009 (pp. 10–14). Japan: Kitakyishu.
- Ulutas, B. (2015). Assessing the number of cells for a cell formation problem. *IFAC Proceedings Volumes*, 48(3), 1122–1127.
- Ulutas, H. B., & Islier, A. A. (2007). A parameter setting for clonal selection algorithm in facility layout problems. In O. Gervasi & M. Gavrilova (Eds.), *LNCS Springer* (pp. 886–899). Berlin: Springer. 4705/2007.
- Ulutas, H. B., & Islier, A. A. (2009). A clonal selection algorithm for dynamic facility layout problems. *Journal of Manufacturing Systems*, 28(4), 123–131.
- Ulutas, B., & Islier, A. A. (2009). *The performance of clonal selection algorithm for cell formation problem compared to other nature based methods*, 4<sup>th</sup> International Conference of Group Technology/Cellular Manufacturing 2009 (GT/CM 2009), 16–18 February 2009 (pp. 15–22). Japan: Kitakyishu.
- Ulutas, B. H., & Kulturel-Konak, S. (2011). A review of clonal selection algorithm and its applications. *Artificial Intelligence Review*, 36(2), 117–138.
- Ulutas, H. B., & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5), 5384–5395.
- Ulutas, H. B., & Kulturel-Konak, S. (2013). Assessing hypermutation operators of clonal selection algorithm for the unequal area facility layout problem. *Engineering Optimization*, 45(3), 375–395.
- Viswanathan, S. (1996). A new approach for solving the P-median problem in group technology. *International Journal of Production Research*, 34(10), 2691–2700.
- Waghodekar, P. H., & Sahu, S. (1984). Machine-component cell formation in group technology, MACE. *International Journal of Production Research*, 22, 937–948.
- Wemmerlov, U., & Hyer, N. (1989). Cellular manufacturing in the US industry: a survey of users. *International Journal of Production Research*, 27(9), 1511–1530.
- Won, Y., & Kim, S. (1997). Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings. *Computers and Industrial Engineering*, 32(1), 207–220.
- Xiangyong, L., Baki, M. F., & Aneja, Y. P. (2010). An ant colony optimization metaheuristic for machine-part cell formation problem. *Computers and Operations Research*, 37, 2071–2081.
- Yang, M. S., & Yang, J. H. (2008). Machine-part cell formation in group technology using a modified ART1 method. *European Journal of Operations Research*, 188(1), 140–152.
- Yasuda, K., Hu, L., & Yin, Y. (2005). A grouping genetic algorithm for the multi-objective cell formation problem. *International Journal of Production Research*, 43(4), 829–853.

- Zolfaghari, S., & Liang, M. (1997). An objective-guided ortho-synapse Hopfield network approach to machine grouping problems. *International Journal of Production Research*, 35(10), 2773–2792.
- Zolfaghari, S., & Liang, M. (2002). Comparative study of simulated annealing, genetic algorithms and tabu search for binary and comprehensive machine-grouping problems. *International Journal of Production Research*, 40, 2141–2158.