Prod. Eng. Res. Devel. (2017) 11:19–28
DOI 10.1007/s11740-016-0706-3

PRODUCTION PROCESS

A hybrid clonal algorithm for the cell formation problem with variant number of cells

Bouchra Karoum[1] · **Bouazza Elbenani**[1]

Received: 24 June 2016 / Accepted: 18 November 2016 / Published online: 28 November 2016

Abstract Cellular manufacturing is an important application of the Group Technology that has been used in several real-world applications such as the electronics industry, offices, structural fabrication, service industries, and hospitals. The manufacturing cell formation problem is considered the first issue faced in designing cellular manufacturing systems in order to overcome difficulties related to multi-product and batch-production systems. The aim is to minimize the inter-cell movements of the parts and maximize the use of the machines. In this paper, a new approach based on the clonal selection algorithm is proposed for solving the problem where the number of cells is not fixed a priori. The approach integrates a local search mechanism to intensify the search of the solutions. To evaluate the effectiveness of the proposed algorithm, a set of 40 benchmark problems is used; the results are then compared to other methods recently developed. The results show that the proposed algorithm performs very well on all test problems since it can reach the best-known solution of 39 benchmark problems (97.5%).

Keywords Cell formation · Grouping efficacy · Local search · Clonal selection · Cellular manufacturing · Affinity

✉ Bouchra Karoum
bouchra.karoum@gmail.com

[1] Research Computer Science Laboratory (LRI), Faculty of Science, Mohammed V University of Rabat, B.P. 1014, Rabat, Morocco

1 Introduction

Due to the permanent changes in the market needs and the rapid development of manufacturing technologies, a Cellular Manufacturing System (CMS) is usually applied to make manufacturing systems more efficient and productive. The Cellular Manufacturing System is one of the most important applications of the Group Technology [1] that attempts to decompose the entire production system into production or manufacturing cells by grouping parts with similar processing needs and identifying the set of machines needed to process these parts. The group of similar parts is denoted part family and the group of machineries used to handle a part family is known as machine cell. The objectives are to overcome difficulties related to multi-product and batch-production systems, reduce setup and throughput times, minimize the material handling costs, improve the quality and the production control, and increment the flexibility, among others [2].

Within the last twenty years, CMS was adopted in different types of industries and applications such as the wooden frames industry [3], the forging industry [4], and the metal-mechanics industry [5]. Also, it was used in ammunition components for defense applications [6], the tractor industry [7], the electronics industry [8], and the health sector [9]. In fact, this kind of manufacturing is effective in those applications that produce quite distinct products.

Among the problems faced in designing a Cellular Manufacturing System, Cell Formation Problem (CFP) is the first and the foremost problem. A previous study showed that the CFP is an NP-hard optimization problem [10], which means that the amount of computation increases exponentially with problem size. Therefore, various methods have been proposed to solve CFP. These

include cluster analysis approaches, graph partitioning approaches, mathematical programming methods and metaheuristic algorithms, among others.

Due to the efficiency of metaheuristic algorithms in solving combinatorial optimization problems, the majority of authors have proposed the use of these algorithms for the resolution of the introduced problem. For example, Goncalves and Resende [11] suggested a new algorithm that combines a local search heuristic with a genetic algorithm for obtaining machine cells and part families. James et al. [12] applied a standard grouping genetic algorithm with a local search mechanism to form machine-part cells. Diaz et al. [13] proposed a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic to obtain lower bounds for the optimal solution of the CFP. Solimanpur and Elmi [14] introduced a nested application of the tabu search approach to solve the problem heuristically. Husseinzadeh et al. [15] presented a new approach based on the Particle Swarm Optimization (PSO) algorithm for solving the CFP. Ying et al. [16] developed a simulated annealing-based metaheuristic with a variable neighbourhood to form part-machine cells. Elbenani et al. [17] combined a local search procedure with a steady-state genetic algorithm to solve CFP, and they also used a destroy and recover strategy to generate new solutions. A grouping version of league championship algorithm has been proposed by Noktehdan et al. [18] for solving the CFP.

The objective of this paper is to present how an adapted metaheuristic algorithm, entitled Clonal Selection Algorithm (CSA), is combined with a local search mechanism for solving the manufacturing cell formation problem, considering the variant number of cells. The proposed method is entitled Hybrid Clonal Selection Algorithm-Cell Formation (HCSA-CF), and it is primarily based on the one introduced by Karoum et al. [19]. One may find similarities between these papers; however, there are significant differences. First, the number of cells in this paper is not fixed a priori, which may affect the intercellular parts movements and the material handling costs. Second, the representation of the solutions in this approach includes the cells selection, which allows the addition or deletion of cells. As a result, the repair process of the infeasible solutions is different from the previous work. Besides, the local search involves a destroy and recover strategy [17] to search more extensively the solutions. Also, the operators of mutation and crossover are applied to intensify the search towards the optimal solution. To evaluate the effectiveness of the proposed algorithm, the HCSA-CF is tested using data sets from the literature; the obtained solutions are then compared with those produced by other methods, taking into consideration a variant number of cells.

The rest of this paper is structured as follows: Sect. 2 describes the CFP; Sect. 3 gives an overview of the HCSA-CF and introduces the improvement carried out on the algorithm to solve the CFP. The results of numerical experiments on a set of benchmark problems are reported in Sect. 4. Finally, concluding remarks are given in Sect. 5.

2 Problem formulation

The cell formation problem is defined by its machine-part incidence matrix A, where each row represents a machine and each column represents a part. The aim is to determine a rearrangement so that the machine utilization within a cell is maximized while the intercellular part movements are minimized.

Figure 1 shows a 7×11 machine-part incidence matrix to a problem with seven machines and eleven parts. The first matrix is an initial matrix where no blocks can be observed directly. The second matrix is the best-known solution of the problem obtained by a partition into four different cells illustrated by the gray blocks. The 1's outside the diagonal blocks are called exceptional elements while the 0's inside the diagonal blocks are called voids. The objective is to minimize the exceptional elements and the voids as much as possible.

Different measures have appeared in the literature to compare the efficiency of the methods. The most used is the grouping efficacy (*Eff*) [20]. The closer the grouping efficacy is to 1, the better the obtained solution, as depicted in Eq. (1).

$$Eff = \frac{(a - a_1^{Out})}{(a + a_0^{In})} \tag{1}$$

where a, total number of entries equal to 1 in the matrix A; a_1^{Out}, number of exceptional elements; a_0^{In}, number of voids.

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

Initial incidence matrix

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

Matrix solution

Fig. 1 Machine-part incidence matrix

The purpose of the grouping efficacy (*Eff*) is to measure the effectiveness of the obtained solution with respect to the constraints of the problem. Where each machine and each part must be assigned to exactly one cell and every cell must include at least one machine and one part.

3 Proposed HCSA-CF

The clonal selection algorithm is a multi-model function optimization algorithm inspired by the biological immune system. It was formally introduced by De Castro et al. [21] as follows:

1. Initialize a population N of individuals randomly;
2. Determine the affinity of each element of the population N;
3. Generate copies of n_1 best highest affinity elements of N proportionally to their affinities;
4. Mutate the generated copies with a rate proportional to their affinities;
5. Add these mutated individuals to the population N and keep n_2 of the best matured individuals as memories of the system;
6. Repeat Steps 2–5 until a certain criterion is met.

This section describes how the clonal selection algorithm is adapted to solve the CFP. The algorithm begins with an initialization phase where the number of parts P, the number of machines M, and the population size are determined. Then, an initial population of feasible solutions is created. Each machine i and part j are assigned randomly to only one cell k in order to satisfy the constraints of the problem. After initialization, the proposed method iterates until a specific number of loops is met. In each iteration, a percentage b of the solutions with the lowest affinity proliferates into several clones according to their affinities. The clones of each selected solution undergo a mutation operator and the one with the highest grouping efficacy (*Eff*) will be used by the crossover operator to produce new solutions. Afterwards, a repair process is performed to fix infeasibilities (empty cell or cell without parts/machines) that may arise during the mutation process. Then, in order to intensify the search towards promising regions, a local search mechanism is applied to two solutions randomly chosen. Finally, a receptor editing process is performed to replace the solutions with the highest affinity by new ones.

3.1 Scheme for coding

In this paper, the encoding scheme fits naturally with the cell formation problem. The chromosome representation consists of three sections: part section; machine section and cell section [12]. The individuals used for solving this problem can be represented as:

$$p_1, p_2, p_3, \ldots, p_P | m_1, m_2, m_3, \ldots, m_M | c_1, c_2, c_3, \ldots, c_K$$

where p_j, denotes what cell part j is assigned to $(j = 1, \ldots, P)$; m_i, denotes what cell machine i is assigned to $(i = 1, \ldots, M)$; and c_k, designates the cell number for cells $1, \ldots, K$.

It is worth mentioning that the lengths of the part and machine sections are both constant for all chromosomes and depend on the size of the problem. The length of the cell section may vary according to the number of cells K into which the parts and machines are divided. In this paper, the number of cells varies from 2 to $Min(M, P)$.

Consider the following example: (4 1 1 2 4 3 3 2 2 4 4| 1 1 2 4 4 3 2| 1 2 3 4). This chromosome represents a possible solution of problem with 11 parts, 7 machines and 4 cells. Where, cell 1 contains parts $\{2, 3\}$ and machines $\{1, 2\}$; cell 2 contains parts $\{4, 8, 9\}$ and machines $\{3, 7\}$; cell 3 contains parts $\{6, 7\}$ and machine $\{6\}$; and finally, cell 4 contains parts $\{1, 5, 10, 11\}$ and machines $\{4, 5\}$.

3.2 Clonal selection

The affinity value is used as a selection mechanism for the current algorithm. The solutions with the lowest affinity in the population will be selected to proliferate into several clones or copies, i.e. the lowest affinity, the higher number of clones. In this paper, the affinity of a solution s is based on its grouping efficacy as depicted in Eq. (2).

$$Affinity(s) = \frac{1}{Eff(s)} \tag{2}$$

The number of clones (*NbClones*) of a solution s depends on its affinity value, the population size and the sum of affinities of the whole population as presented in Eq. (3) [21].

$$NbClones = \frac{Affinity(s)}{\sum_{t=1}^{popsize} Affinity(t)} \times popsize \tag{3}$$

3.3 Mutation

The mutation operator tends to avoid trapping in a local optimum by searching in a new neighborhood. The one used in this paper is the exchange mutation or the pair wise mutation. Given a solution s, let u and v two positions randomly chosen from the parts and machines sections. A neighbor of s is obtained by interchanging the cells of two machines, two parts or machine-part in positions u and v.

To clarify further, take the following example:

$s = (4\ 1\ 1\ 2\ 4\ 3\ 3\ 2\ 2\ 4\ 4|\ 1\ 1\ 2\ 4\ 4\ 3\ 2|\ 1\ 2\ 3\ 4)$

The neighbor s' is obtained by interchanging the cells of part 6 and machine 4. Where part 6 is moved from cell 3 to cell 4, and machine 4 is moved from cell 4 to cell 3.

$s' = (4\ 1\ 1\ 2\ 4\ \underline{\mathbf{4}}\ 3\ 2\ 2\ 4\ 4|\ 1\ 1\ 2\ \underline{\mathbf{3}}\ 4\ 3\ 2|\ 1\ 2\ 3\ 4)$

If the grouping efficacy value of the mutated solution is better than that of the original, then it is stored for use in the crossover operator phase.

3.4 Crossover

The crossover operator combines information from two parents such that the two children have a 'resemblance' to each parent. In the proposed method, a uniform crossover is applied where two solutions, called parents, are randomly selected from the already mutated and stored solutions.

Consider the two parent solutions:

Parent 1: $(P_1^1, \ldots, P_p^1, M_1^1, \ldots, M_M^1)$

Parent 2: $(P_1^2, \ldots, P_p^2, M_1^2, \ldots, M_M^2)$

A crossover mask vector of bits having $(P + M)$ elements is generated: $(S_1, \ldots, S_P, S_{(P+1)}, \ldots, S_{(P+M)})$.

The offspring solutions $(OP_1^r, \ldots, OP_p^r, OM_1^r, \ldots, OM_M^r)$, $r = 1, 2$, are specified as follows:

$$For\ j = 1, \ldots, P,\ If:$$

$$S_j = \begin{cases} 1, & then\ OP_j^1 = P_j^1\ and\ OP_j^2 = P_j^2 \\ 0, & then\ OP_j^1 = P_j^2\ and\ OP_j^2 = P_j^1 \end{cases}$$

$$For\ i = 1, \ldots, M,\ If:$$

$$S_{P+i} = \begin{cases} 1, & then\ OM_i^1 = M_i^1\ and\ OM_i^2 = M_i^2 \\ 0, & then\ OM_i^1 = M_i^2\ and\ OM_i^2 = M_i^1 \end{cases}$$

To fix infeasibilities that may occur from the genetic operators, a repair heuristic similar to the one proposed by Brown and Sumichrast [22], is performed. This heuristic involves removing the empty cells (cells without parts or machines), then assigning the orphaned parts or machines to cells for which they are best adapted. The novelty in this paper is the way of determining the best cell as described below.

- If the displaced item is a part j, it will be attributed to the cell c with the most machines required by this product, as depicted in Eq. (4).

$$c_j = \max_{k=1,\ldots,K} \left\{ \frac{a_{1j}(k)}{a_{0j}(k) + a_{.j}} \right\} \tag{4}$$

  where $a_{1j}(k)$, the number of machines in cell k that are processing part j; $a_{0j}(k)$, the number of machines in cell k that are not processing part j; $a_{.j}$, the number of machines that are processing part j.

- Likewise in Eq. (5), if the displaced item is a machine i, then the cell c containing the majority of the parts that necessitate this machine will take the orphaned machine.

$$c_i = \max_{k=1,\ldots,K} \left\{ \frac{a_{i1}(k)}{a_{i0}(k) + a_{i.}} \right\} \tag{5}$$

  where $a_{i1}(k)$, the number of parts in cell k that are processed by machine i; $a_{i0}(k)$, the number of parts in cell k that are not processed by machine i; $a_{i.}$, the number of parts that are processed by machine i.

3.5 Local search

In order to intensify the search and improve the quality of the solutions, the proposed clonal selection algorithm is combined with a local search mechanism. The adopted mechanism is inspired from the one introduced by Elbenani et al. [17].

Based on the initial set of machine groups of the incoming solution, each part is assigned to the cell where it is best suited using Eq. (4). If the modified solution is better than the original solution, the original solution is replaced and the process will restart by the assignment of the machines this time. This approach iterates by reassignment of parts, then reassignment of machines until the quality of the new solution does not exceed the quality of the last solution.

However, if no modification is possible in the original solution, the destroy and recover strategy, already described in [17], is applied to slightly change the current solution.

4 Computational results and comparison to other approaches

In this section, the proposed algorithm is tested on 40 benchmark problems collected from the literature. The size of these problems ranges from 7 parts and 5 machines to 100 parts and 40 machines in order to evaluate the scalability of the algorithm.

This algorithm is coded using Java, and the numerical tests are completed on a Personal Computer equipped with Intel ® Core™ i7 clock at 2.50 GHz and 8 GB of RAM.

4.1 Parameter settings

In this paper, three parameters can affect the performance of the proposed method: the population size (*popsize*), the maximum number of iterations, and the percentage b of the selected antibodies for cloning process. Among these

parameters, the population size and the maximum number of iterations depend on the size of the problem.

For problems with small size, the population size and the maximum number of iterations can be set to small values. On the contrary, those two parameters should be higher for problems with a large size. In this paper, the maximum number of iterations is set to 100 iterations.

A statistical study based on the analysis of variance (ANOVA), using SPSS software, is designed to determine the significant factors that influence the efficacy of the solutions. According to related research, three possible levels for factor *b* (0.2, 0.25, 0.3) and four possible levels for factor *popsize* (20, 30, 40, 50) were considered. A test of significance at 95% confidence level is employed to spot the significance of these factors. A factor can affect the grouping efficacy if it gets a high F (Fisher's test) value and low P (or Sig) value.

Since the large size problems are difficult to solve, we choose a problem with 40 parts and 24 machines (P26 in Table 2) for the statistical analysis. Table 1 presents the

Table 1 ANOVA table

Source	DF[a]	SS	MS	F	P
popsize	3	1.820	0.607	3.520	0.018
b	2	0.328	0.164	0.951	0.389
popsize ∗ *b*	6	3.004	0.501	2.906	0.012
Error	108	18.611	0.172		
Total	120	276231.190			

[a] *DF* Degree of freedom, *SS* sum of squares, *MS* mean square, *F* Fisher's test, *P* probability

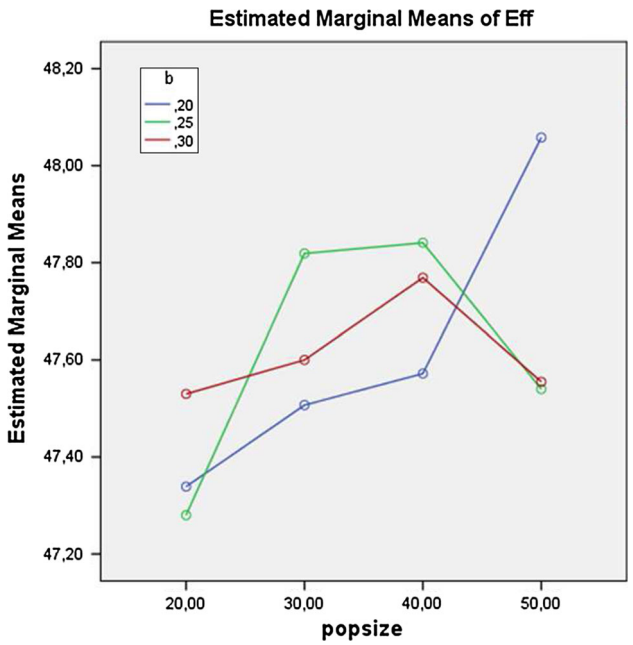

Fig. 2 Interaction plot *popsize*b*

obtained results. As can be seen in this table, the probability *P* of *popsize* is <0.05 ($P_{popsize} = 0.018$); we conclude that the factor *popsize* affects on the efficacy of the solution. Additionally, $P_{popsize*b} < 0.05$ demonstrates that there is an interaction between the population size and the percentage *b*. Figure 2 presents this interaction and shows that the proper values of *b* and *popsize* are 0.2 (20%) and 50, respectively.

4.2 Results

The proposed HCSA-CF is tested on 40 benchmark problems. 10 independent runs of the algorithm are carried out for each benchmark problem with the parameters mentioned before. Table 2 illustrates the source and the size of each benchmark problem. The minimum, average, and maximum solutions reached by our proposed algorithm, and the computation time required in seconds are also presented in this table, respectively. In addition, Table 2 presents the best-known solution of each test problem. In this paper, the percentage of the grouping efficacy is used to evaluate the performance of the method.

Firstly, it should be noted that our method does not accept singleton, i.e. a cell containing only one part or one machine. Therefore, these types of solution are not accepted as the best-known solutions in this paper.

As can be seen in Table 2, the proposed HCSA-CF achieves the best-known solutions in 39 benchmark problems out of the existing 40 test problems. Among these 39 problems, our method attains the best-known solution of 25 problems (P1 to P13, P20, P22 to P24, P31, P33, P34, P36 and P40) throughout the 10 runs of the algorithm ($Min.Sol. = Ave.Sol. = Max.Sol.$), which confirm the performance of the introduced HCSA-CF. Concerning the residual problems, the average solutions obtained are very close to the best known solutions.

Regarding the computational time needed to find the best solution, the first 11 problems are easy to solve since they require less than a half second of computational time. The proposed method can also reach the best-known solution for 14 of the larger problems (P12 to P15, P17 to P19, and P36 to P42) in <5 s. Unfortunately, the solution time becomes larger for the remaining problems because of their sizes (*M* and *P*). Nevertheless, our method generates very good solutions in a reasonable computational time.

To confirm the effectiveness of the proposed method, the obtained results are compared with the results of several methods recently developed for solving the CFP where the number of cells is not fixed a priori, as depicted in Table 3. The comparator methods are hybrid grouping genetic algorithm (HGGA) [12], hybrid grouping

Table 2 Computational results on 40 benchmark problems

No.	Source	M	P	Min. Sol.	Avg. Sol.	Max. Sol.	Time(s)	Best known sol.
P1	King and Nakornachai [23]	5	7	82.35	82.35	82.35	0.02	82.35
P2	Waghodekar and Sahu [24]	5	7	69.57	69.57	69.57	0.02	69.57
P3	Seifoddini [25]	5	18	79.59	79.59	79.59	0.07	79.59
P4	Kusiak and Cho [26]	6	8	76.92	76.92	76.92	0.03	76.92
P5	Kusiak and Chow [27]	7	11	60.87	60.87	60.87	0.07	60.87
P6	Boctor [28]	7	11	70.83	70.83	70.83	0.09	70.83
P7	Seifoddini and Wolfe [29]	8	12	69.44	69.44	69.44	0.12	69.44
P8	Chandrasekharan and Rajagopalon [30]	8	20	85.25	85.25	85.25	0.28	85.25
P9	Chandrasekharan and Rajagopalon [31]	8	20	58.72	58.72	58.72	0.20	58.72
P10	Mosier and Taube [32]	10	10	75.00	75.00	75.00	0.22	75.00
P11	Chan and Milner [33]	10	15	92.00	92.00	92.00	0.40	92.00
P12	Askin and Subramanian [34]	14	24	72.06	72.06	72.06	1.65	72.06
P13	Stanfel [35]	14	24	71.83	71.83	71.83	1.56	71.83
P14	McCormick et al. [36]	16	24	52.69	53.20	53.26	2.41	53.26
P15	Srinivasan et al. [37]	16	30	69.4	69.50	69.53	2.97	69.53
P16	King [38]	16	43	57.32	57.32	57.53	8.12	57.53
P17	Carrie [39]	18	24	56.31	57.38	57.73	4.28	57.73
P18	Mosier and Taube [40]	20	20	42.54	42.92	43.45	4.66	43.45
P19	Kumar et al. [41]	20	23	49.6	50.44	50.81	4.13	50.81
P20	Carrie [39]	20	35	77.91	77.91	77.91	11.60	77.91
P21	Boe and Cheng [42]	20	35	57.51	57.93	57.98	9.81	57.98
P22	Chandrasekharan and Rajagopalan [43]	24	40	100.0	100.0	100.0	25.47	100.0
P23	Chandrasekharan and Rajagopalan [43]	24	40	85.11	85.11	85.11	30.21	85.11
P24	Chandrasekharan and Rajagopalan [43]	24	40	73.51	73.51	73.51	31.83	73.51
P25	Chandrasekharan and Rajagopalan [43]	24	40	52.8	53.04	53.29	36.36	53.29
P26	Chandrasekharan and Rajagopalan [43]	24	40	47.95	48.11	48.95	36.27	48.95
P27	McCormick et al. [36]	27	27	54.45	54.77	54.82	16.57	54.82
P28	Kumar and Vannelli [44]	30	41	62.14	62.66	63.31	74.24	63.31
P29	Stanfel [35]	30	50	58.7	59.4	59.77	98.56	59.77
P30	McCormick et al. [36]	37	53	60.54	60.62	60.64	123.71	60.64
P31	Chandrasekharan and Rajagopalan [45]	40	100	84.03	84.03	84.03	690.81	84.03
P32	Chandrasekharan and Rajagopalan [43]	16	30	52.17	52.26	52.29	2.46	52.29
P33	Chandrasekharan and Rajagopalan [43]	16	30	63.04	63.04	63.04	2.40	63.04
P34	Chandrasekharan and Rajagopalan [43]	16	30	68.38	68.38	68.38	3.52	68.38
P35	Chandrasekharan and Rajagopalan [43]	16	30	49.29	49.55	50.00	2.92	50.00
P36	Chandrasekharan and Rajagopalan [43]	16	30	72.73	72.73	72.73	3.46	72.73
P37	Chandrasekharan and Rajagopalan [43]	16	30	77.31	77.31	77.31	2.88	77.31
P38	Chandrasekharan and Rajagopalan [43]	16	30	73.24	73.24	73.24	3.18	73.24
P39	Chandrasekharan and Rajagopalan [43]	20	35	77.3	77.3	77.3	10.95	77.3
P40	Chandrasekharan and Rajagopalan [43]	24	40	72.37	72.37	72.37	28.30	72.37

differential evolution algorithm (HGDE) [46], hybrid particle swarm optimization (HGBPSO) [15], and grouping league championship algorithm (GLCA) [18].

The comparison with the other methods shows that the proposed HCSA-CF outperforms HGGA, HGBPSO and HGDE in P15 and P36 in terms of quality solution. Concerning the GLCA method, the singleton appears in the solution obtained for P36, unlike our method. In problems P18, P32, P33 and P35, HCSA-CF found better results than HGGA and surpassed HGBPSO in P18. There is only one problem P29 for which HCSA-CF couldn't reach the best-known solution found in the literature. Regarding problem P14, the optimal solution is already reached using an exact method [47] and it is equal to 53.26% (the same value

Table 3 Performance of the proposed method compared to other approaches

No.	HGGA			HGDE			HGBPSO			GLCA			HCSA-CF			Best known sol.
	Min.	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	
P1	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**	**82.35**
P2	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**	**69.57**
P3	**79.59**	**79.59**	**79.59**	**79.59**	**79.59**	**79.59**	**79.59**	**79.59**	**79.59**	**79.59**	79.84	80.85[a]	**79.59**	**79.59**	**79.59**	**79.59**
P4	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**	**76.92**
P5	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**	**60.87**
P6	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**	**70.83**
P7	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**	**69.44**
P8	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**	**85.25**
P9	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**	**58.72**
P10	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**
P11	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**	**92**
P12	**72.06**	**72.06**	**72.06**	**72.06**	**72.06**	**72.06**	**72.06**	**72.06**	**72.06**	73.53	73.53	73.53[a]	**72.06**	**72.06**	**72.06**	**72.06**
P13	**71.83**	**71.83**	**71.83**	71.62	71.79	**71.83**	71.62	71.81	**71.83**	**71.83**	**71.83**	**71.83**	**71.83**	**71.83**	**71.83**	**71.83**
P14	52.75	52.75	52.75	53.33	53.37	53.41	53.41	53.41	53.41	53.33	53.39	53.41	52.69	53.20	**53.26**	**53.26**
P15	68.99	68.99	68.99	68.99	68.99	68.99	68.99	68.99	68.99	68.99	68.99	68.99	69.4	69.5	**69.53**	**69.53**
P16	**57.53**	**57.53**	**57.53**	57.43	57.51	**57.53**	**57.53**	**57.5**3	**57.53**	57.31	57.46	**57.53**	57.32	57.32	**57.53**	**57.53**
P17	57.43	57.7	**57.73**	**57.73**	**57.73**	**57.73**	57.43	57.7	**57.73**	**57.73**	**57.73**	**57.73**	56.31	57.38	**57.73**	**57.73**
P18	42.74	42.93	43.18	42.42	42.95	**43.45**	42.74	43	43.26	42.55	42.78	**43.45**	42.54	42.92	**43.45**	**43.45**
P19	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	**50.81**	49.6	50.44	**50.81**	**50.81**
P20	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**	**77.91**
P21	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	**57.98**	57.51	57.93	**57.98**	**57.98**
P22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
P23	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**	**85.11**
P24	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**	**73.51**
P25	53.15	53.27	**53.29**	**53.29**	**53.29**	**53.29**	53.21	53.28	**53.29**	**53.29**	**53.29**	**53.29**	52.8	53.04	**53.29**	**53.29**
P26	48.55	48.66	**48.95**	48.61	48.75	**48.95**	48.61	48.72	**48.95**	–	–	–	47.95	48.11	**48.95**	**48.95**
P27	53.41	53.84	54.02	54.02	54.02	54.02	54.13	54.48	**54.82**	54.69	54.79	**54.82**	54.45	54.77	**54.82**	**54.82**
P28	62.24	62.74	**63.31**	62.42	62.85	**63.31**	62.24	62.8	**63.31**	62	62.95	**63.31**	62.14	62.66	**63.31**	**63.31**
P29	59.77	59.77	59.77	59.77	59.77	59.77	59.77	59.77	59.77	59.44	59.59	59.77	58.7	59.4	59.77	60.12
P30	60.48	60.57	**60.64**	60.48	60.59	**60.64**	60.4	60.58	**60.64**	58.03	59.48	**60.64**	60.54	60.62	**60.64**	**60.64**
P31	**84.03**	**84.03**	**84.03**	–	–	–	–	–	–	–	–	–	**84.03**	**84.03**	**84.03**	**84.03**
P32	52.05	52.05	52.05	52.07	52.17	**52.29**	51.43	51.83	**52.29**	52.07	52.18	**52.29**	52.17	52.26	**52.29**	**52.29**
P33	62.99	62.99	62.99	62.29	63.03	**63.04**	62.6	62.77	**63.04**	**63.04**	**63.04**	**63.04**	**63.04**	**63.04**	**63.04**	**63.04**
P34	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**	**68.38**

Table 3 continued

No.	HGGA			HGDE			HGBPSO			GLCA			HCSA-CF			Best known sol.
	Min.	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	*Min.*	*Ave.*	*Max.*	
P35	49.29	49.6	49.65	49.62	49.75	**50**	49.28	49.72	**50**	49.64	49.96	**50**	49.29	49.55	**50**	**50**
P36	72.37	72.37	72.37	72.37	72.37	72.37	72.37	72.37	72.37	72.37	72.37	**72.73**[a]	**72.73**	**72.73**	**72.73**	**72.73**
P37	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**	**77.31**
P38	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**	**73.24**
P39	76.51	76.6	**77.3**	76.51	77	**77.3**	76.51	76.86	**77.3**	–	–	–	**77.3**	**77.3**	**77.3**	**77.3**
P40	**72.37**	**72.37**	**72.37**	**72.37**	**72.37**	**72.37**	**72.37**	**72.37**	**72.37**	–	–	–	**72.37**	**72.37**	**72.37**	**72.37**

Bold values indicate the solutions equal to the best-known solutions

[a] The solution where singleton appears

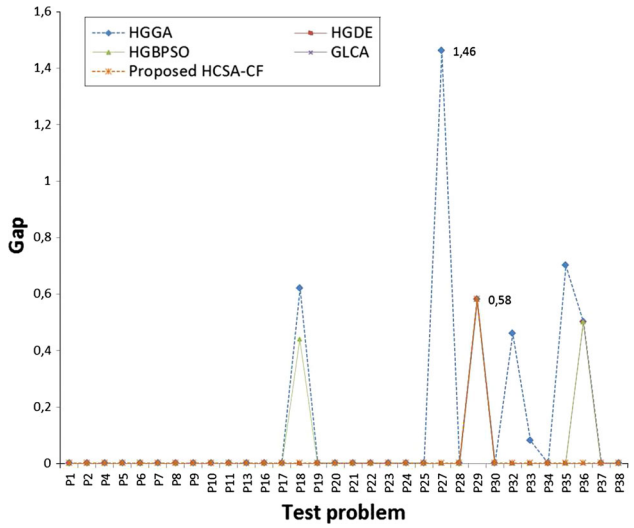

Fig. 3 The gaps comparison

Table 4 Comparison among all algorithms in terms of finding best solution

	HGGA	HGDE	HGBPSO	GLCA	Proposed HCSA-CF
Number of best solutions	31 out of 40	35 out of 39	34 out of 39	30 out of 35	39 out of 40
Percentage of reach best solutions %	77.5	89.7	87.1	85.7	97.5

attained by our method), which is conflicting with the one obtained by other methods.

Table 4 summarizes the results of the comparison among the comparator algorithms and the proposed algorithm in terms of finding the best solution and shows that our method dominates the others by reaching the best-known solutions of 97.5% of benchmark problems.

Figure 3 depicts the gap comparison between our method and the comparator algorithms. The gap value is calculated using Eq. (6). This value is calculated for the test problems resolved by all the algorithms. The solutions with singletons are not considered.

$$Gap = \frac{(Best\ known\ sol. - Max.\ sol.)}{Best\ known\ sol.} \times 100 \quad (6)$$

As illustrated in Fig. 3, GLCA and HCSA-CF methods have gap equal to 0.58% in problem P29 only. HGDE algorithm follows with gaps in two problems with the highest one for problem P29 equal to 0.58%. The HGBPSO method is placed the third with gaps in three problems and the highest one is for problem 29 equal to 0.58%. Finally, the HGGA algorithm has gaps in seven problems with the

highest one for problem 27 being equal to 1.46%. Therefore, the obtained results confirm the excellent performance of the proposed HCSA-CF.

5 Conclusion

The cell formation problem is one of the most important issues faced in designing cellular manufacturing systems. In this paper, a new approach is proposed for solving the manufacturing cell formation problem based on minimizing the number of exceptional elements and voids. It is worth mentioning that the fact of having an unfixed number of cells makes the resolution of the problem more difficult.

The proposed HCSA-CF is evaluated using 40 benchmark problems collected from the literature. The obtained results confirm its effectiveness since it can reach 39 out of 40 problems (97.5%) in reasonable computational time. The quality of the solutions is compared with those produced by the previous well-known methods: hybrid grouping genetic algorithm (HGGA), hybrid grouping differential evolution algorithm (HGDE), hybrid particle swarm optimization (HGBPSO), and grouping league championship algorithm (GLCA). Consequently, we found that our method outperforms the other methods.

For future research, an extension of the approach could be applied to other variants of the cell formation problem having additional factors such as the real-time manufacturing data, the alternative routing process, the production volume, and the machine capacity.

References

1. Wemmerlov U, Hyer NL (1986) Procedures for the part family/machine group identification problem in cellular manufacturing. J Oper Manag 6:125–147
2. Wemmerlov U, Hyer NL (1989) Cellular manufacturing in the US industry: a survey of users. Int J Prod Res 27:1511–1530
3. Dinis-Carvalho J, Alves AC, Sousa RM (2014) Moving from jobshop to production cells without losing flexibility: a case study from the wooden frames industry. S Afr J Ind Eng 25:212–225
4. Hung K-T, Maleki H (2014) Applying group technology to the forging industry. Prod Plan Control Manag Oper 25:134–148
5. Johnson DJ, Wemmerlv U (2004) Why does cell implementation stop? Factors influencing cell penetration in manufacturing plants. Prod Oper Manag 13:272–289
6. Pattanaik LN, Sharma BP (2008) Implementing lean manufacturing with cellular layout: a case study. Int J Adv Manuf Technol 42:772–779
7. Shayan E, Sobhanallahi A (2002) Productivity gains by cellular manufacturing. Prod Plan Control 13:507–516
8. Levasseur GA, Helms MM, Zink AA (1995) A conversion from a functional to a cellular manufacturing layout at Steward, Inc. Prod Invent Manag J 36:37–42
9. Landsbergis PA, Cahill J, Schnall P (1999) The impact of lean production and related new systems of work organization on worker health. J Occup Health Psychol 4:108–130
10. Dimopoulos C, Zalzala AM (2000) Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. IEEE Trans Evolut Comput 4:93–113
11. Goncalves J, Resende M (2004) An evolutionary algorithm for manufacturing cell formation. Comput Ind Eng 47:247–273
12. James TL, Brown EC, Keeling KB (2007) A hybrid grouping genetic algorithm for the cell formation problem. Comput Oper Res 34:2059–2079
13. Diaz JA, Luna D, Luna R (2012) A GRASP heuristic for the manufacturing cell. Top 20:679–706
14. Solimanpur M, Elmi A (2013) A tabu search approach for cell scheduling problem with makespan criterion. Int J Prod Econ 141:639–645
15. Husseinzadeh Kashan A, Karimi B, Noktehdan A (2014) A novel discrete particle swarm optimization algorithm for the manufacturing cell formation problem. Int J Adv Manuf Technol 73:1543–1556
16. Ying KC, Lin SW, Lu CC (2011) Cell formation using a simulated annealing algorithm with variable neighbourhood. Eur J Ind Eng 5:22–42
17. Elbenani B, Ferland JA, Bellemare J (2012) Genetic algorithm and large neighbourhood search to solve the cell formation problem. Expert Syst Appl 39:2408–2414
18. Noktehdan A, Seyedhosseini S, Saidi-Mehrabad M (2015) A metaheuristic algorithm for the manufacturing cell formation problem based on grouping efficacy. Int J Adv Manuf Technol 82:25–37. doi:10.1007/s00170-015-7052-z
19. Karoum B, Elbenani B, El Imrani AA (2016) Clonal selection algorithm for the cell formation problem. In: El Oualkadi A et al (eds) Proceedings of the Mediterranean conference on information and communication technologies 2015, Lecture Notes in Electrical Engineering, vol 380, pp 319–326
20. Kumar CS, Chandrasekharan MP (1990) Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. Int J Prod Res 28:233–243
21. De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evolut Comput 6:239–251
22. Brown EC, Sumichrast RT (2001) CF-GGA: a grouping genetic algorithm for the cell formation problem. Int J Prod Res 39:3651–3669
23. King JR, Nakornchai V (1982) Machine-component group formation in group technology: review and extension. Int J Prod Res 20:117–133
24. Waghodekar PH, Sahu S (1984) Machine-component cell formation in group technology: MACE. Int J Prod Res 22:937–948
25. Seifoddini H (1989) A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. Int J Prod Res 27:1161–1165
26. Kusiak A, Cho M (1992) Similarity coefficient algorithms for solving the group technology problem. J Manuf Syst 30:2633–2646
27. Kusiak A, Chow WS (1987) Efficient solving of the group technology problem. J Manuf Syst 6:117–124
28. Boctor FF (1991) A linear formulation of the machine-part cell formation problem. Int J Prod Res 29:343–356
29. Seifoddini H, Wolfe PM (1986) Application of the similarity coefficient method in group technology. IIE Trans 18:271–277
30. Chandrasekharan MP, Rajagopalan R (1986) MODROC: an extension of rank order clustering for group technology. Int J Prod Res 24:1221–1233
31. Chandrasekharan MP, Rajagopalan R (1986) An ideal seed nonhierarchical clustering algorithm for cellular manufacturing. Int J Prod Res 24:451–463

32. Mosier C, Taube L (1985) The facets of group technology and their impacts on implementation—a state-of-the-art survey. Omega 13:381–391
33. Chan HM, Milner DA (1982) Direct clustering algorithm for group formation in cellular manufacture. J Manuf Syst 1:65–75
34. Asktn RG, Subramantan SP (1987) A cost-based heuristic for group technology configuration. Int J Prod Res 25:101–113
35. Stanfel LE (1985) Machine clustering for economic production. Eng Costs Prod Econ 9:73–81
36. McCormick WT (1972) Problem decomposition and data reorganization by a clustering technique. Oper Res 20:993–1009
37. Srinivasan G, Narendran TT, Mahadevan B (1990) An assignment model for the part-families problem in group technology. Int J Prod Res 28:145–152
38. King JR (1980) Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. Int J Prod Res 18:213–232
39. Carrie AS (1973) Numerical taxonomy applied to group technology and plant layout. Int J Prod Res 11:399–416
40. Mosier C, Taube L (1985) Weighted similarity measure heuristics for the group technology machine clustering problem. Omega 13:577–579
41. Kumar KR, Kusiak A, Vannelli A (1986) Grouping of parts and components in flexible manufacturing systems. Eur J Oper Res 24:387–397
42. Boe WJ, Cheng CH (1991) A close neighbour algorithm for designing cellular manufacturing systems. Int J Prod Res 29:2097–2116
43. Chandrasekharan MP, Rajagopalan R (1989) GROUPABIL1TY: an analysis of the properties of binary data matrices for group technology. Int J Prod Res 27:1035–1052
44. Kumar KR, Vannelli A (1987) Strategic subcontracting for efficient disaggregated manufacturing. Int J Prod Res 25:1715–1728
45. Chandrasekharan MP, Rajagopalan R (1987) ZODIAC—an algorithm for concurrent formation of part-families and machine-cells. Int J Prod Res 25:835–850
46. Noktehdan A, Karimi B, Husseinzadeh KA (2010) A differential evolution algorithm for the manufacturing cell formation problem using group based operators. Expert Syst Appl 37:4822–4829
47. Thanh LT, Ferland JA, Elbenani B, Dinh Thuc N, Hien Nguyen V (2015) A computational study of hybrid approaches of meta-heuristic algorithms for the cell formation problem. J Oper Res Soc 67:20–36