Prod. Eng. Res. Devel. (2017) 11:545–556
DOI 10.1007/s11740-017-0751-6

PRODUCTION MANAGEMENT

A clonal selection algorithm for the generalized cell formation problem considering machine reliability and alternative routings

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

Received: 1 March 2017 / Accepted: 28 June 2017 / Published online: 11 July 2017

Abstract The cell formation is the first step in the design of Cellular Manufacturing systems. It consists of grouping parts with similar processing needs into cells and identifying the set of machines needed to process these parts. The aim is to minimize the material handling costs and maximize the use of the machines. In this paper, the machine reliability and the alternative process routings are taken into account to form the production cells. The presence of these factors in addition to the production volume, operation sequence and production time makes the problem more realistic but also more complex. Most authors solve this kind of problems by mathematical programming approaches that require large amounts of computational efforts. Therefore, a modified version of the Clonal Selection Algorithm is introduced and a local search mechanism is adopted in this paper. The obtained results are compared with those of the Branch and Bound (B&B) method using LINGO software. The comparison reveals the effectiveness and the efficiency of the proposed method in terms of both solution quality and computation time required.

Keywords Cell formation problem · Clonal selection algorithm · Material handling costs · Alternative routings · Machine reliability · Production volume · Production time

✉ 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

Cellular Manufacturing is an important application of the Group Technology that groups parts with similar manufacturing characteristics into parts families and assigned them to machine cells. The objective is to upgrade the quality and the production control, reduce the material handling costs, increment the flexibility, decrease the setup and throughput times, and others more [1]. The first issue faced in designing a Cellular Manufacturing system is the Cell Formation Problem (CFP). A previous study showed that the CFP is an NP-hard optimization problem [2].

Several researchers estimate that each part has a single process routing with specific operations, and all machines are 100% reliable. But, in the real manufacturing environment, a part can be produced through different routes with a certain number of operations. These operations can be performed on several alternative machines, ensuring flexibility to the manufacturing system design. The possibility of alternative routes transforms the CFP into a Generalized Cell Formation Problem (GCFP) [3]. The main objective of this problem is to minimize the material handling costs, particularly the intercellular part movements costs. Such costs represent the charges of moving the parts of their current cells in order to be processed on different machines located in other cells. Therefore, machines are essential components in manufacturing systems, but they may be exposed to breakdowns due to usage. This is difficult to overcome as quickly as the production system requires even with the alternative process routings. These breakdowns can badly affect the system performance measures and lead to scheduling problems, which decrease the productivity of the entire manufacturing operation. Consequently, the consideration of machine reliability is important for the design of the Cellular Manufacturing system in order

to improve the performance of the system [4, 5]. However, a limited number of authors have studied the reduction of the machine breakdown costs, which generally includes the machine repairing costs, production suspension costs, and capacity lost costs [6].

The aim of this paper is to propose a new method, entitled Hybrid Clonal Selection Algorithm-Generalized Cell Formation (HCSA-GCF), that minimizes the costs of intercellular part movements and machine breakdowns with consideration of the production volume and the processing time of each operation. The proposed HCSA-GCF is tested on a set of benchmark problems. The obtained solutions are compared with those provided by LINGO software. Our main objective is to improve the best-found solutions and solve large-scale problems in a reasonable computational time.

The rest of this article is organized as follows: Sect. 2 presents a literature review of the GCFP; Sect. 3 describes the GCFP and provides a numerical example; Sect. 4 introduces the modifications applied to the standard Clonal Selection Algorithm for solving the GCFP. The experimental results are reported in Sect. 5. Finally, Sect. 6 concludes.

2 Related work

A limited number of authors have addressed machine breakdowns or reliability issues in GCFP resolution with the presence of production volume, production sequence and production time. Apparently, the integration of these factors into one model makes the CFP more complex but also more realistic.

Das et al. [7] presented a multi-objective model for the design of Cellular Manufacturing systems with integration of the machine reliability. They assumed that each part has alternative process routings and each machine has a number of similar copies. The purpose of their model is to maximize the system reliability and minimize the total costs. An important issue regarding this model is that it defines the reliability of the whole system depending on the reliability of the individual machines.

Jabal Ameli and Arkat [6] proposed a new binary integer programming model for the GCFP with consideration of the alternative routes and the reliability of machines. They solved the model using a mathematical programming method. Also, they added a new cost term measuring the machine reliability cost based on the Mean Time Between Failure (MTBF) and the production time. These authors have contributed in another work [4], in which they indicated notable negative aspects of some traditional approaches. Such approaches have considered the reliability of the manufacturing system as a function of its elements. Besides, the authors have presented a new approach to model the effects of the machine unreliability and to confirm the efficacy of the new term of the machine breakdowns cost.

Due to the non-polynomial nature of GCFP, the resolution of large-scale instances using classical optimization methods requires a large amount of computational efforts. As a result, some authors proposed metaheuristic algorithms for solving the introduced problem.

Various authors were inspired by the mathematical model of Jabal Ameli and Arkat [6] for solving the GCFP. For example, Jabal Ameli et al. [8] developed three algorithms, namely: Simulated Annealing, Genetic Algorithm, and Memetic Algorithm to solve this model. The comparison between those algorithms and the Branch and Bound algorithm reveals the outperformance of the Memetic Algorithm in term of the quality of the solution, plus the efficiency of the Simulated Annealing algorithm regarding the execution time required. The authors aimed to maximize the reliability of the manufacturing systems and to minimize the intercellular part movements costs where the number of cells is fixed in advance. Nevertheless, other practical production factors such as the duplication of the machines and the intracellular part movements were not considered in their paper. Chung et al. [9] adopted the same model and suggested a Tabu Search algorithm to solve the problem. They used a similarity coefficient to produce good initial solutions. In their method, the number of cells is not prescribed beforehand. It increases by 1 as long as solution improvement is observed. Such mechanism may influence the runtime of the algorithm. The comparison with LINGO software reveals the superiority of their method. However, the intracellular part movements were not considered in their study. Jouzdani et al. [10] added the intracellular part movements and the setup costs in the mathematical model; then they presented a modified version of the Simulated Annealing algorithm to solve this model. Their objective was to minimize the material handling and the setup costs and to improve the reliability of the machines. The possibility of multiple machines of identical type was not considered by the authors. The obtained results showed that the proposed method is promising. For future research, the authors pointed out in the importance of using the number of machines as model variables.

Liu et al. [11] developed a three-stage heuristic algorithm for solving the GCFP with consideration of the production volume and material handling costs. They first form the temporary machine group plan and select the appropriate process routing of each part with respect to the material handling costs; then they configure the regular manufacturing cells according to the selected process routings. However, their proposed model is not efficient to some problems with machine reliability or operation time constraints. Kao and Lin [12] introduced a Particle Swarm Optimization

(PSO) based two-stage procedure for the GCFP. In the first step, the machines are grouped into different cells using a discrete version of PSO; then the part routing is assigned to each machine cell with an objective to minimize the number of exceptional parts outside cells. The obtained results were promising, but the authors have not addressed several practical production factors such as the material handling costs, the reliability of the machines, and the production time. Solimanpur et al. [13] introduced a solution approach based on Ant Colony Optimization to minimize the intercellular part movements with consideration of operations sequence and production volume. In their paper, the authors emphasized the importance of those factors for calculating intercellular part movements; but, they didn't take into account the constraints related to machines such as its capacity and reliability. Mohammadi and Forghani [14] proposed a hybrid method combining the genetic algorithm and dynamic programming to solve a new bi-objective cell formation problem. They sought to reduce the total dissimilarity between parts and to minimize the machine investment cost with consideration of alternative process routings and machine duplication.

Table 1 lists some methods proposed for solving the GCFP with the presence of other production factors to explicitly indicate the various problems addressed by the authors. As can be seen, some important researches have not considered the sequence of operations and the production volume in the calculation of intercellular movements while it is directly influenced by those parameters. In addition, a limited number of authors have considered the reliability of machines.

The essence of Cell Formation approaches is to minimize even eliminate the intercellular part movements costs. This can merely be attained by using multiple identical machines. However, the duplication of machines involves large capital investment that influences the production cost. Therefore, the proposed manufacturing cell formation approach must ensure an optimization amongst these. For this reason, our model depends on the one introduced by Jabal Ameli and Arkat [6].

Table 1 List of research papers that considered alternative process routings and other various production factors

References	AR	BD	OS	InterC	IntraC	MC	MR	MM	PT	PV	CS	Methods
Adenso-Diaz et al. [15]	x										x	TS
Yin and Yasuda [16]	x		x	x			x	x	x	x		HA
Mukattash et al. [17]	x							x	x			HA
Jayaswal and Adil [18]	x		x	x				x		x	x	SA
Kim et al. [19]	x		x	x								HA
Hu and Yasuda [20]	x		x	x					x	x		GA
Spiliopoulos and Sofianopoulos [21]	x		x	x				x		x	x	HA
Ameli et al. [8]	x		x	x			x		x	x	x	SA, GA, MA
Sormaz and Rajaraman [22]	x			x							x	SS
Safaei et al. [23]	x		x	x				x				SA
Wu et al. [24]	x											SA
Cao et al. [25]	x			x					x			GA
Solimanpur et al. [13]	x		x	x						x	x	ACO
Liu et al. [11]	x			x	x					x	x	HA
Chung et al. [9]	x		x	x			x		x	x	x	TS
Kao and Lin [12]	x	x										PSO
Tavakkoli-Moghaddam et al. [26]	x		x	x		x		x	x	x	x	SS
Ozcelik and Sarac [27]	x										x	GA
Khaksar-Haghani et al. [28]	x		x	x	x		x	x	x	x		GA
Vin and Delchambre [29]	x			x	x					x		GA
Saeidi et al. [30]	x		x	x				x	x	x		GA
Jouzdani et al. [10]	x		x	x	x		x		x	x	x	SA
Deep and Singh [31]	x			x	x		x			x		GA
Mohammadi and Forghani [14]	x					x		x	x	x	x	GA and DP

AR alternative routings, *BD* binary data, *OS* operation sequence, *InterC* intercellular movement cost, *IntraC* intracellular movement cost, *MC* machine capacity, *MR* machine reliability, *MM* multiple machine, *PT* processing time/cost, *PV* production volume/demand, *CS* cell size, *HA* heuristic algorithm, *SA* simulated annealing, *MA* memetic algorithm, *ACO* ant colony optimization, *TS* Tabu search, *PSO* particle swarm optimization, *SS* scatter search, *DP* dynamic programming

3 Problem description

In this section, the GCFP with machine reliability and alternative process routings considerations is presented as a 0–1 integer programming model.

3.1 Assumptions

The introduced problem is investigated under the following assumptions:

- The total number of cells is known;
- The lower bound and the upper bound for the number of machines in each cell are known;
- Each part type has one or more processing routes, but only one route can be chosen;
- The operations of each processing route are performed on different machines based on a given order. Moreover, the sequence of operations is important in the calculation of intercellular material handling costs, since it can influence the number of times the part has to move between machines of different cells or between machines within the same cell;
- The processing time of each operation for every part type on each machine is determined;
- One machine of each type is used. The multiple identical machines are not taking into account in this paper;
- The demand or volume for every part type is known and deterministic;
- The machine reliability is calculated using the Mean Time Between Failure (MTBF) concept introduced by Jabal Ameli et al. [6].

3.2 Mathematical model

3.2.1 Parameters

m: Number of machines.
n: Number of parts.
c: Number of cells.
r: Total number of routings.
P_i: Production volume for part i.
q_i: Number of routings for part i.
L_l: Lower bound of the number of machines in cell l.
U_l: Upper bound of the number of machines in cell l.
K_{ij}: Number of machines in routing j of part i.
$\{u_{ij}^{(1)}, u_{ij}^{(2)}, \ldots, u_{ij}^{(k_{ij})}\}$: Machines index in routing j of part i.
T_{ik}: Processing time for part i on machine k.
B_k: Breakdown cost for machine k.
A_{ij}: Intercellular movement cost for part i in routing j.
$MTBF_k$: Mean time between failure of machine k.

3.2.2 Decision variables

$$Y_{kl} = \begin{cases} 1, & \text{if machine } k \text{ locates in cell } l \\ 0, & \text{otherwise.} \end{cases}$$

$$Z_{ij} = \begin{cases} 1, & \text{if routing } j \text{ of part } i \text{ is selected} \\ 0, & \text{otherwise.} \end{cases}$$

$$X_{ijklsl} = \begin{cases} & \text{if routing } j \text{ of part } i \text{ is selected,} \\ 1 & \text{machine } k \text{ locates in cell } l \\ & \text{and machine } s \text{ do not locate in cell } l. \\ 0, & \text{otherwise.} \end{cases}$$

3.2.3 Objective function

The objective is to minimize the cost of the intercellular part movements and maximize the machine reliability by reducing the breakdown cost of machines as shown in Eq. (1).

$$\min TC = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{K_{ij}-1}\sum_{l=1}^{c} A_{ij}P_i X_{ij(u_{ij}^{(k)})l(u_{ij}^{(k+1)})l} + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{K_{ij}} Z_{ij}\frac{P_i T_{i(u_{ij}^{(k)})} B_{(u_{ij}^{(k)})}}{MTBF_{(u_{ij}^{(k)})}} \tag{1}$$

Subject to:

$$\sum_{j=1}^{q_i} Z_{ij} = 1 \quad i = 1, 2, \ldots, n \tag{2}$$

$$\sum_{k=1}^{m} Y_{kl} \le U_l \quad l = 1, 2, \ldots, c \tag{3}$$

$$\sum_{k=1}^{m} Y_{kl} \ge L_l \quad l = 1, 2, \ldots, c \tag{4}$$

$$\sum_{l=1}^{c} Y_{kl} = 1 \quad k = 1, 2, \ldots, m \tag{5}$$

$$X_{ijklsl} \le Z_{ij} \tag{6}$$

$$X_{ijklsl} \le Y_{kl} \tag{7}$$

$$X_{ijklsl} \le (1 - Y_{sl}) \tag{8}$$

Table 2 Numerical example data

Sequence (processing time (min))	Parts (volume)																	
Machines	P1(75)			P2(130)			P3(110)		P4(145)		P5(110)		P6(105)		P7(140)			P8(115)	
	R1	R2	R3	R1	R2	R3	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R3	R4	R1	R2
M1	1(5)			1(4)			1(4)	1(4)	1(5)	1(5)		1(4)		1(4)	1(5)			1(5)	1(4)	
M2		1(5)	1(5)		1(5)	1(5)					1(3)		1(5)			1(3)	1(3)	2(3)		1(4)
M3							2(3)	2(3)	2(3)	2(3)										
M4	2(3)			2(4)				3(3)	3(5)	3(5)		2(4)			2(4)		2(4)			
M5	3(4)		2(4)	3(3)	2(3)	2(3)	3(4)			4(4)					3(3)		3(3)			
M6		2(5)														2(4)		3(4)	2(3)	2(4)
M7											2(5)	3(5)	2(5)	2(5)						
M8		3(4)		4(4)		3(4)	4(3)	4(3)	4(3)	5(4)			3(5)	3(5)			4(5)			
M9	4(5)		3(5)		3(3)						3(5)	4(5)			4(5)	3(5)		4(5)		
Inter_C	375	375	0	1300	0	650	1100	0	0	1450	1100	550	525	0	700	0	700	700	575	0

Table 3 Machine reliability for the numerical example

Machine	MTBF (h)	Breakdown cost
1	90	900
2	51	2000
3	73	2000
4	60	1600
5	76	1500
6	62	1800
7	71	1400
8	58	1700
9	65	1500

$$Z_{ij} + Y_{kl} + (1 - Y_{sl}) - X_{ijklsl} \leq 2 \tag{9}$$

$$Z_{ij}, Y_{kl}, X_{ijklsl} \in \{0, 1\} \tag{10}$$

The first part of the objective function (Eq. 1) represents the cost of moving the parts between different cells. The second part calculates the breakdown cost of the machines. This is obtained by dividing the production time by the MTBF, then multiplying this quantity by the unit machine breakdown cost. The constraint (Eq. 2) imposes that just one process routing must be selected for each part. The inequalities (Eq. 3) and (Eq. 4) impose the upper and lower limits for the number of machines in each cell, respectively. The constraint (Eq. 5) indicates that each machine will be assigned to exactly one cell. The constraints (Eqs. 6–8) guarantee that if one of the primary binary variables has a zero value, then their corresponding new variables will take a zero value as well. The constraint (Eq. 9) ensures that if all of the primary variables take unit values, then their corresponding new variables take unit values as well. Finally, the constraint (Eq. 10) denotes that the variables are binary.

3.3 Numerical example

To clarify further the introduced GCFP, a numerical example, proposed by Bhide et al. [32] and slightly modified by Jouzdani et al. [10] and Alhourani [33], is used. This example is received from a gear factory. It contains 8 parts, 9 machines, 20 process routings, and two cells. The minimum number of machines needed in each cell is two, and the maximum number of machines permitted in each cell is equal to six.

Tables 2 and 3 provide the data for the numerical example. Table 2 shows the part-machine incidence matrix. The first and the second rows of this table represent the demands (production volume) and the alternative routes for each part. The last row displays the intercellular movement cost of the parts in each route. The numbers in the table are in the form 'operation sequence (operation time)'; i.e.,

Table 4 The results from LINGO for the numerical example

		Part (Route)							
		P1(R3)	P5(R1)	P7(R2)	P2(R1)	P3(R1)	P4(R1)	P6(R2)	P8(R1)
Machines	M2	1	1	1					
	M7		2					2	
	M9	3	3	3					
	M1				1	1	1	1	1
	M3					2	2		
	M4				2		3		
	M5	2			3	3			
	M6			2					2
	M8				4	4	4	3	

it shows the sequence of operations regarding the process of the parts along the routes and the processing time of the corresponding operation. Table 3 presents the machines, their corresponding MTBF, and the breakdown costs, respectively.

The proposed model is solved using the LINGO software. Table 4 presents the solution obtained with breakdown cost equal to 284, 530.7 and nil intercellular movement cost. The first cell contains machines {2, 7, 9}; and parts {1, 5, 7}. The second cell includes machines {1, 3, 4, 5, 6, 8}; and parts {2, 3, 4, 6, 8}. Only one route is selected for each part. For example, part 1 is processed using the sequence of operations along the route 3, part 5 is assigned to route 1 and so on.

4 Proposed Clonal Selection Algorithm

The Clonal Selection Algorithm is a metaheuristic algorithm inspired by the artificial immune system where the antigen represents the optimization problem and the antibody corresponds to a possible solution of the problem. This algorithm was officially presented by De Castro and Von Zuben [34]. In our previous works, we have proposed different versions of the algorithm to solve the CFP with a fixed and variable number of cells [35–37].

In this paper, the consideration of the machine reliability and the alternative process routings makes the problem more realistic but also more complex. This section demonstrates how the Clonal Selection Algorithm is adapted for solving the GCFP. An overview of the proposed HCSA-GCF is depicted in Fig. 1. The algorithm starts with a generalization of feasible solutions (antibodies) that satisfy the constraints of the problem. The affinity value of every solution is calculated. Subsequently, exact copies (clones) of the solutions with the highest affinity are produced proportionally to their affinities. Then, new solutions are developed by mutating these copies to change them slightly. Afterward, a crossover operator is applied to the new solutions and a repair process is used to fix infeasibilities that may appear during the mutation process. In order to improve the quality of the solutions and avoid trapping in local optimum, a local search method is applied to two solutions randomly selected. Finally, a receptor editing process is used as a diversification phase of the algorithm by replacing the solutions with the lowest affinity in the population by new ones.

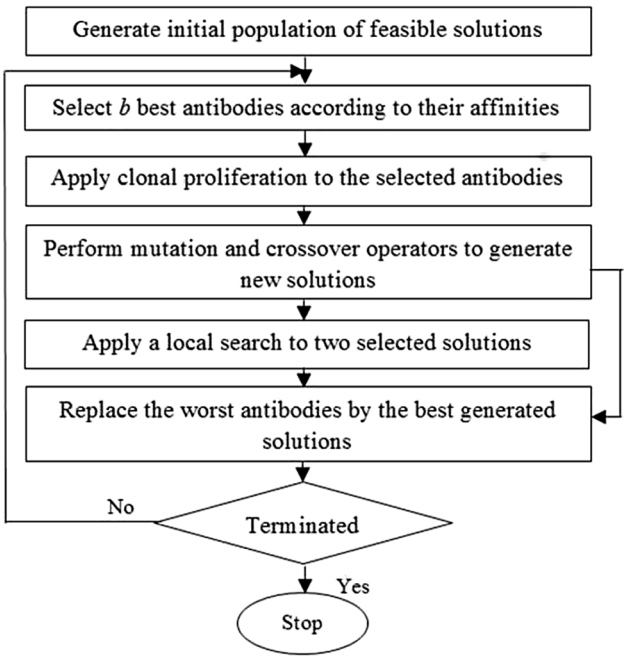

Fig. 1 Flowchart of the proposed algorithm

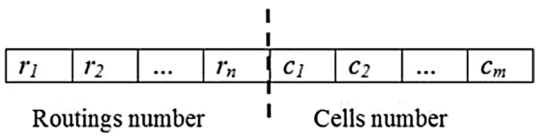

Fig. 2 Solution representation

4.1 Solution representation

The solution structure has two sections; the first for the selected process routing of each part and the second for the cell of every machine. Each solution is represented

as a vector with a length of $n + m$, where n is the number of parts and m is the number of machines as presented in Fig. 2.

In this representation, r_i is the routing number of part i, and c_j is the cell number of machine j. Since the part families do not affect the objective function, they are not considered as a part of the model. After finding machine cells, each part is assigned to the cell with the highest number of machines processing this part [8].

For example, the solution obtained before by LINGO for the numerical example (see Table 4) can be presented as (3, 1, 1, 1, 1, 2, 2, 1|2, 1, 2, 2, 2, 2, 1, 2, 1).

4.2 Clonal proliferation

The number of clones of a solution s is calculated based on its affinity value, the population size and the sum of affinities of the whole population as depicted by Eq. 11 [34].

$$NbClones = \frac{Affinity(s)}{\sum_{t=1}^{popsize} Affinity(t)} \times popsize. \tag{11}$$

The affinity value of a solution s is based on its objective function (TC) as illustrated by Eq. (12). Therefore, the solutions with the highest affinity will be proliferated into many copies or clones.

$$Affinity(s) = \frac{1}{TC}. \tag{12}$$

4.3 Mutation and crossover

In this paper, the mutation is performed to the cloned solutions by altering an element (part's route or machine's cell) of the original solution.

To understand better, consider the following example:

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

s' is obtained by changing the processing route of part 2 from route 1 to route 3.

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

The mutated solutions will then undergo a uniform crossover to combine the elements of every two parent solutions and generate new solutions.

Take the two parent solutions:
Parent 1: $(r_1^1, \ldots, r_n^1, c_1^1, \ldots, c_m^1)$
Parent 2: $(r_1^2, \ldots, r_n^2, c_1^2, \ldots, c_m^2)$

A crossover mask vector of bits having $(n + m)$ elements is generated:

$(e_1, \ldots, e_n, e_{(n+1)}, \ldots, e_{(n+m)})$.

The offspring solutions $(or_1^v, \ldots, or_n^v, oc_1^v, \ldots, oc_m^v)$, $v = 1, 2$, are specified as follows:

For $i = 1, \ldots, n$, *If*:

$$e_i = \begin{cases} 1, & then\ or_i^1 = r_i^1\ and\ or_i^2 = r_i^2 \\ 0, & then\ or_i^1 = r_i^2\ and\ or_i^2 = r_i^1 \end{cases}$$

For $k = 1, \ldots, m$, *If*:

$$e_{n+k} = \begin{cases} 1, & then\ oc_k^1 = c_k^1\ and\ oc_k^2 = c_k^2 \\ 0, & then\ oc_k^1 = c_k^2\ and\ oc_k^2 = c_k^1 \end{cases}$$

Let:
Parent 1: (3, 1, 1, 1, 1, 2, 2, 1|2, 1, 2, 2, 2, 2, 1, 2, 1)
Parent 2: (2, 2, 1, 2, 1, 1, 4, 2|1, 1, 2, 1, 2, 1, 1, 2, 1)
Mask: (0, 1, 1, 0, 1, 0, 1, 1|0, 1, 0, 0, 1, 1, 1, 0, 1)
Then, the generated offspring solutions in this case are:
Offspring 1: (2, 1, 1, 2, 1, 1, 2, 1|1, 1, 2, 1, 2, 2, 1, 2, 1)
Offspring 2: (3, 2, 1, 1, 1, 2, 4, 2|2, 1, 2, 2, 2, 1, 1, 2, 1)

The modifications made by those operators could produce infeasible solutions. Consequently, a repair heuristic is applied to fix them. This heuristic implicates adding or removing machines from the infeasible cells with respect to the upper and lower bounds for the number of machines in every cell.

4.4 Local search

In addition to the characteristics of the Clonal Selection Algorithm, we used a local search mechanism to improve the quality of the solutions and to intensify the search towards new regions. This mechanism is applied to two solutions randomly selected from the whole population.

Based on the processing routes chosen for the parts, we modify the machine cells. Each machine will be assigned to the cell that improves the objective function. If the modified solution is better than the current solution, the modified solution replaces the current one. Afterwards, the process restarts by assignment of the processing routes based on the machine groups. This approach is repeated by reassignment of machines, then reassignment of routes until the quality of the obtained solution does not surpass the quality of the last solution. It is noteworthy that the order of choosing the element (machine or route) to reassign is arbitrary.

5 Computational results

To assess the efficacy of the proposed method, a set of suitable test problems is prepared. However, only a limited amount of research has dealt with the machine breakdown costs and the intercellular movement costs. Consequently, the data are randomly generated. In this paper, 16 test problems are used as shown in Table 5. Among them, two (#5, #7) are collected from the literature. The remaining

Table 5 Data of test problems

No.	Source	Size ($m \times n \times r$)	Randomly generated data
1	Kusiak [3]	4 × 5 × 11	BC, MTBF, PV, IC and TP
2	Moon and Chi [38]	6 × 6 × 13	BC, MTBF, PV, IC and TP
3	Sankaran and Kasilingam [39]	6 × 10 × 20	BC, MTBF and TP
4	Lee et al. [40]	8 × 13 × 26	BC, MTBF, IC and TP
5	Alhourani [33]	9 × 8 × 20	–
6	Oum Kim et al. [41]	10 × 10 × 25	BC, MTBF, IC
7	Alhourani [33]	10 × 16 × 32	–
8	Alhourani [42]	12 × 12 × 20	BC, MTBF, PV, IC
9	Sofianopoulou [43]	12 × 20 × 26	BC, MTBF, PV, IC and TP
10	Sofianopoulou [43]	14 × 20 × 26	BC, MTBF, PV, IC and TP
11	Jabal Ameli and Arkat [6]	17 × 30 × 63	IC
12	Sofianopoulou [43]	18 × 30 × 59	BC, MTBF, PV, IC and TP
13	Nagi et al. [44]	20 × 20 × 51	BC, MTBF, IC
14	Won and Kim [45]	26 × 28 × 71	BC, MTBF, PV, IC and TP
15	Lee et al. [40]	30 × 40 × 89	BC, MTBF, IC and TP
16	Hu and Yasuda [46]	30 × 70 × 149	BC, MTBF, PV, IC and TP

Table 6 Design factors and their levels

Parameters	Number of levels	Values
popsize	4	20, 30, 40, 50
b	3	0.2, 0.25, 0.3

Table 7 ANOVA table

Source	DF[a]	SS	MS	F	P
popsize	3	4.90E+8	1.63E+8	1.75	0.161
b	2	1.08E+9	5.42E+8	5.82	0.004
popsize × *b*	6	1.40E+9	2.34E+8	2.51	0.026
Error	108	1.00E+10	9.32E+7		
Total	120	3.05E+15			

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

problems are prepared by adding randomly generated data like machine breakdown cost (BC), cost of intercellular part movement in each route (IC), mean time between failure (MTBF), production volume (PV), and production time (PT). Table 5 gives a detailed data of each new test problem.

The proposed method 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.

5.1 Parameter settings

There are three control parameters must be handled: the population size (*popsize*), the maximum number of iterations and the percentage *b* for cloning process. Amongst these, the number of iterations and the population size rely on the size of the test problem. For problems with small (large) size, the values of those two parameters may be smaller (higher). The maximum number of iterations is set to 500 iterations. The most interesting parameter is the percentage *b* that has a great effect on the final solution.

Depending on related researches, three possible levels for factor *b* and four possible levels for factor *popsize* were considered as shown in Table 6. In this paper, we applied a statistical study to define the significant factors that affect the efficacy of the solutions. This study is based on ANOVA: analysis of variance, using SPSS software. The factor with the highest F (Fisher's test) and lowest P (Probability) influence the objective function (TC). A confidence level equal to 95% is used to recognize the important factors.

To ensure that the values of the selected parameters are appropriate for the proposed method. We choose the large size benchmark problem #16 (see Table 5) for the statistical analysis. 3 × 4 full factorial experiment is designed for this problem. Ten replicates of a complete factorial experiment are run within it. As a result, 120 experiments are performed.

Table 7 presents the obtained results. As seen in the table, the probability P of *b* is less than 0.05 ($P_b = 0.004$); we deduce that the factor *b* influences the efficacy of the solution (*TC*). Additionally, $P_{popsize*b} < 0.05$ proves that there is an interaction between the population size and the percentage *b*. To determine the proper levels of the factors, we should begin with the factor with the highest

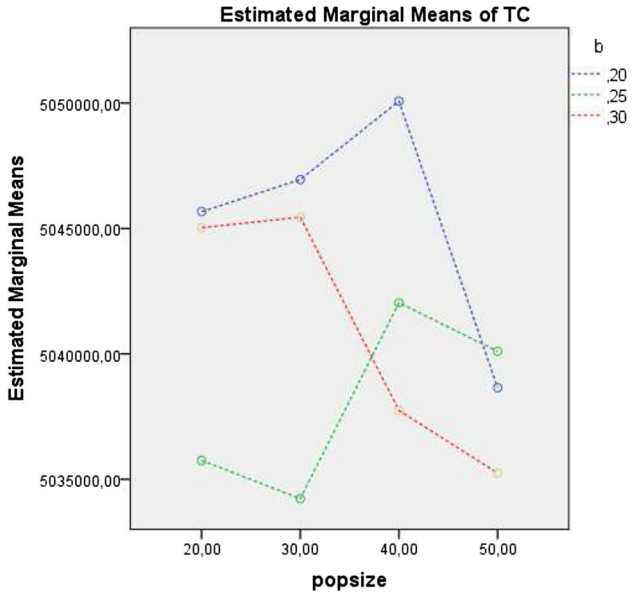

Fig. 3 Interaction plot *popsize*b* of large size problem

F values. According to Fig. 3, the proper values are: $b = 0.25$ and $popsize = 30$.

5.2 Results

Due to its stochastic features, ten independent runs of the HCSA-GCF are performed for each test problem with the parameters mentioned before. The obtained results are compared with the optimal solutions resulted from the LINGO 16.0 software using the B&B method. The computational results are summarized in Table 8. For each test problem, the total intercellular cost (*TIC*), the total breakdown cost (*TBC*), the total cost ($TC = TIC + TBC$), and the computational time required in seconds (*CPU(s)*), obtained from both the LINGO software and our method, are presented in this table, respectively.

As seen in the table, the proposed method could reach the global optimum for 14 instance problems (#1 to #13, #15) in less than 3 s; while LINGO requires a large computational time to solve the majority of these problems. The obtained results confirm the ability of HCSA-GCF to solve the problem in a reasonable computational time.

Regarding the instance #14, LINGO was not able to find the global optimum solution after more than 72 h of running. However, our proposed method attained the same value as LINGO in less than 2 s as depicted in Table 8.

As for instance #16, owing to its large size, LINGO could not find the optimal solution after more than 72 h of running. It reached a value equal to 5038604. On the other hand, the proposed HCSA-GCF attained a better solution in less than 8 s, which outperforms the solution of LINGO in terms of solution quality and running time required.

In order to demonstrate the contribution of the local search mechanism in this paper, we compare the proposed HCSA-GCF with the results obtained without using the local search mechanism CSA-GCF. Table 9 summarizes the obtained solutions.

Table 8 Results comparison of the proposed method and optimal solutions of LINGO 16.0

No.	LINGO software				Proposed method HCSA-GCF			
	TIC	*TBC*	*TC*	*CPU (s)*	*TIC*	*TBC*	*TC*	*CPU (s)*
1	5250	54,609.9	**59859.9**[a]	0.53	5250	54,609.9	**59,859.9**	0.04
2	0	68,260.24	**68260.24**[a]	0.23	0	68,260.24	**68,260.24**	0.05
3	60,000	1,955,554	**2015554**[a]	0.26	60,000	1,955,553.8	**2,015,554**	0.11
4	0	258,234	**258234**[a]	0.84	0	258,234	**258,234**	0.1
5	0	284,530.7	**284530.7**[a]	0.47	0	284,530.7	**284,530.7**	0.14
6	2000	50,436.48	**52436.48**[a]	0.78	2000	50,436.48	**52,436.48**	0.14
7	6000	2341.32	**8341.32**[a]	1.47	6000	2341.32	**8341.32**	0.2
8	3350	71,828.74	**75178.74**[a]	31.03	3350	71,828.74	**75,178.74**	0.27
9	21,580	80,482.3	**102062.3**[a]	32.06	21,580	80,482.3	**102,062.3**	0.56
10	4840	80,249.89	**85089.89**[a]	51.37	4840	80,249.89	**85,089.89**	0.52
11	40,125	43,458.38	**83583.38**[a]	68	40,125	43,458.38	**83,583.38**	0.6
12	191,400	993,056.5	**1184456**[a]	346.34	191,400	993,056.5	**1,184,456**	1.45
13	0	443.02	**443.02**[a]	65.26	0	443.02	**443.02**	0.49
14	56,000	873,198.3	**929198.3**	–	56,000	873,198.3	**929,198.3**	1.63
15	6800	1,127,687	**1134487**[a]	428	6800	1,127,687	**1,134,487**	1.71
16	672,080	4,366,524	5038604	–	641,330	4,389,076.19	**5,030,406.19**	7.35

[a]Global optimum

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

Table 9 Results comparison of the proposed method with and without local search

No.	CSA-GCF				HCSA-GCF			
	TIC	*TBC*	*TC*	*CPU (s)*	*TIC*	*TBC*	*TC*	*CPU (s)*
1	5250	54,609.9	**59,859.9**	0.03	5250	54,609.9	**59,859.9**	0.04
2	0	68,260.24	**68,260.24**	0.03	0	68,260.24	**68,260.24**	0.05
3	60,000	1,955,554	**2,015,554**	0.03	60,000	1,955,553.8	**2,015,554**	0.11
4	0	258,234	**258,234**	0.05	0	258,234	**258,234**	0.1
5	0	284,530.7	**284,530.7**	0.01	0	284,530.7	**284,530.7**	0.14
6	2000	50,436.48	**52,436.48**	0.04	2000	50,436.48	**52,436.48**	0.14
7	6000	2341.32	**8341.32**	0.07	6000	2341.32	**8341.32**	0.2
8	3350	71,828.74	**75,178.74**	0.07	3350	71,828.74	**75,178.74**	0.27
9	21,580	80,482.3	**102,062.3**	0.15	21,580	80,482.3	**102,062.3**	0.56
10	4840	80,249.89	**85,089.89**	0.17	4840	80,249.89	**85,089.89**	0.52
11	40,125	43,458.38	**83,583.38**	0.14	40,125	43,458.38	**83,583.38**	0.6
12	191,400	993,056.5	**1,184,456**	0.22	191,400	993,056.5	**1,184,456**	1.45
13	0	443.42	443.42	0.12	0	443.02	**443.02**	0.49
14	58,200	873,854.13	932,054.13	0.3	56,000	873,198.3	**929,198.3**	1.63
15	29,600	1123,968.69	1,153,568.69	0.67	6800	1,127,687	**1,134,487**	1.71
16	678,000	4354,675.8	5,032,675.8	1.8	641,330	4,389,076.19	**5,030,406.19**	7.35

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

Table 10 Comparison among *LB*, LINGO, CSA-GCF and HCSA-GCF

No.	*LB*	LINGO	CSA-GCF	HCSA-GCF
1	**59,859.9**	**59,859.9**	**59,859.9**	**59,859.9**
2	**68,260.24**	**68,260.24**	**68,260.24**	**68,260.24**
3	**2,015,554**	**2,015,554**	**2,015,554**	**2,015,554**
4	**258,234**	**258,234**	**258,234**	**258,234**
5	**284,530.7**	**284,530.7**	**284,530.7**	**284,530.7**
6	**52,436.48**	**52,436.48**	**52,436.48**	**52,436.48**
7	**8341.32**	**8341.32**	**8341.32**	**8341.32**
8	**75,178.74**	**75,178.74**	**75,178.74**	**75,178.74**
9	**102,062.3**	**102,062.3**	**102,062.3**	**102,062.3**
10	**85,089.89**	**85,089.89**	**85,089.89**	**85,089.89**
11	**83,583.38**	**83,583.38**	**83,583.38**	**83,583.38**
12	**1,184,456**	**1,184,456**	**1,184,456**	**1,184,456**
13	**443.02**	**443.02**	443.42	**443.02**
14	**920,612.1**	929,198.3	932,054.13	929,198.3
15	**1,134,487**	**1,134,487**	1,153,568.69	**1,134,487**
16	**4,526,388**	5,038,604	5,032,675.8	5,030,406.19

Bold values indicate the solutions equal to *LB*

As shown in this table, the CSA-GCF reached the best-known solution for 12 test problems. This is indicated by marking in bold the values. Regarding the remaining test problems (#13 to #16), the obtained results are close to the best-known solutions. However, to enhance the performance of the CSA-GCF, we integrated the local search mechanism.

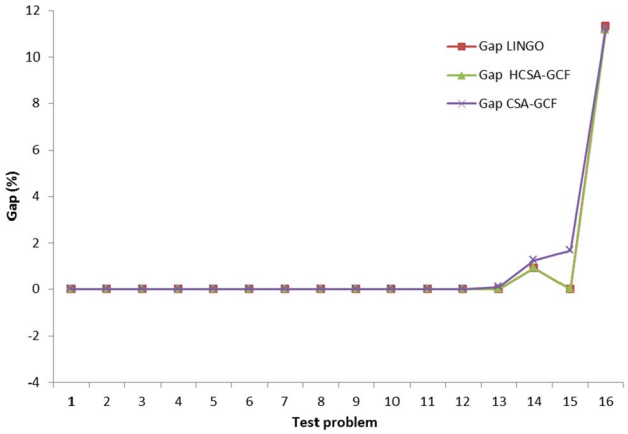

Fig. 4 Gap comparison of CSA-GCF, HCSA-GCF and LINGO

To prove the effectiveness of the proposed method, the gap value is calculated based on the lower bound found by LINGO for each test problem using Eq. (13). The lower bound denoted by *LB* is used as the optimality criteria of the solutions. Table 10 and Fig. 4 summarize the results of the three methods.

$$Gap = \frac{Solution - LB}{LB} \times 100\% \tag{13}$$

The results show that the proposed HCSA-GCF is ranked first having gaps in two instances problems with the highest one is in #16 equal to 11.13%. LINGO is in the second rank with 11.31% gap in test problem #16. CSA-GCF

has gaps in four problems with the highest one is in problem #16 equal to 11.19%. In conclusion, the proposed method is more efficient than both CSA-GCF and LINGO regarding solving the GCFP. Unfortunately, we could not compare our results with other methods since the majority of the authors generate the test problems randomly to fit their adapted models.

6 Conclusion

This paper proposed a new method to solve the Generalized Cell Formation Problem with the presence of production volume, production sequence, machine reliability, and alternative process routings. Considering all these factors makes the problem more realistic, but also more complex. Inspired by the effectiveness of the metaheuristic, we adopted the Clonal Selection Algorithm to solve the problem. To the best of our knowledge, this algorithm is applied for the first time to solve this kind of problems.

The proposed method is evaluated using a set of benchmark problems. The obtained results are compared with those produced by Branch and Bound algorithm, using the software LINGO. The comparison results justify the outperformance of the presented method in terms of quality of the solutions and the required computational time.

The main contributions of our work are:

- The presence of alternative process routings and machine reliability in addition to other factors makes the GCFP more realistic and interesting for real-world applications.
- The proposed HCSA-GCF was improved to solve large scale problems effectively.
- The combination of the Clonal Selection Algorithm with a local search mechanism to improve the quality of the solutions.
- The resolution of the GCFP in a very reasonable computational time comparing with the software LINGO.
- The size of the problems ranges from small to large in order to evaluate the scalability of the proposed algorithm.

Unfortunately, our work still has no industrial applications, but we look forward to apply it to one of these applications.

For future research, the proposed model can be extended by integrating other factors such as intracellular movement costs, human resource assignment costs, and others more. Also, we aim to apply this algorithm for resolution of real-world manufacturing systems.

References

1. Wemmerlov U, Hyer NL (1989) Cellular manufacturing in the US industry: a survey of users. Int J Prod Res 27:1511–1530
2. Dimopoulos C, Zalzala AM (2000) Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. IEEE Trans Evol Comput 4:93–113
3. Kusiak A (1987) The generalized group technology concept. Int J Prod Res 25:561–569
4. Jabal Ameli MS, Arkat J, Barzinpour F (2008) Modelling the effects of machine breakdowns in the generalized cell formation problem. Int J Adv Manuf Technol 39:838–850
5. D'Addona D, Teti R (2011) Queuing network modelling techniques for response time enhancement in electronics assembly. Int J Comput Aided Eng Technol 3:399–413
6. Jabal Ameli MS, Arkat J (2008) Cell formation with alternative process routings and machine reliability consideration. Int J Adv Manuf Technol 35:761–768
7. Das K, Lashkari RS, Sengupta S (2007) Reliability consideration in the design and analysis of cellular manufacturing systems. Int J Prod Econ 105:243–262
8. Jabal Ameli MS, Arkat J, Sakri MS (2008) Applying metaheuristics in the generalized cell formation problem considering machine reliability. J Chin Inst Ind Eng 25:261–274
9. Chung SH, Wu TH, Chang CC (2011) An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. Comput Ind Eng 60:7–15
10. Jouzdani J, Barzinpour F, Shafia MA, Fathian M (2014) Applying simulated annealing to a generalized cell formation problem considering alternative routings and machine reliability. Asia Pac J Oper Res. doi:10.1142/S0217595914500213
11. Liu CG, Yin Y, Yasuda K, Lian J (2010) A heuristic algorithm for cell formation problems with consideration of multiple production factors. Int J Adv Manuf Technol 6:12011213
12. Kao Y, Lin CH (2012) A PSO-based approach to cell formation problems with alternative process routings. Int J Prod Res 50(15):40754089
13. Solimanpur M, Saeedi S, Mahdavi I (2010) Solving cell formation problem in cellular manufacturing using ant-colony-based optimization. Int J Adv Manuf Technol 50:1135–1144
14. Mohammadi M, Forghani K (2017) A hybrid method based on genetic algorithm and dynamic programming for solving a bi-objective cell formation problem considering alternative process routings and machine duplication. Appl Soft Comput 53:97–110
15. Adenso-Diaz BL, Ozano S, Racero J, Guerrero F (2001) Machine cell formation in generalized group technology. Comput Ind Eng 41:227–240
16. Yin Y, Yasuda K (2002) Manufacturing cells' design in consideration of various production factors. Int J Prod Res 40:885–906
17. Mukattash AM, Adil MB, Tahboub KK (2002) Heuristic approaches for part assignment in cell formation. Comput Ind Eng 42:329–341
18. Jayaswal S, Adil GK (2004) Efficient algorithm for cell formation with sequence data, machine replications and alternative process routings. Int J Prod Res 42:2419–2433
19. Kim CO, Baek JG, Baek JK (2004) A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences. Int J Prod Res 42(18):3911–3927
20. Hu L, Yasuda K (2006) Minimising material handling cost in cell formation with alternative processing routes by grouping genetic algorithm. Int J Prod Res 44(11):2133–2167
21. Spiliopoulos K, Sofianopoulou S (2007) Manufacturing cells design with alternative routings in generalized group technology:

reducing the complexity of the solution space. Int J Prod Res 45:1355–1367
22. Sormaz DN, Rajaraman SN (2008) Problem space search algorithm for manufacturing cell formation with alternative process plans. Int J Prod Res 46(2):345–369
23. Safaei N, Saidi-Mehrabad M, Jabal-Ameli MS (2008) A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. Eur J Oper Res 185:563–592
24. Wu TH, Chung SH, Chang CC (2009) Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings. Expert Syst Appl 36:3652–3661
25. Cao D, Defersha FM, Chen M (2009) Grouping operations in cellular manufacturing considering alternative routings and the impact of run length on product quality. Int J Prod Res 47(4):989–1013
26. Tavakkoli-Moghaddam R, Ranjbar-Bourani M, Amin GR, Siadat A (2012) A cell formation problem considering machine utilization and alternative process routes by Scatter search. J Intell Manuf 23:1127–1139
27. Ozcelik F, Sarac T (2012) A genetic algorithm extended modified sub-gradient algorithm for cell formation problem with alternative routings. Int J Prod Res 50:4025–4037
28. Khaksar-Haghani F, Kia R, Mahdavi I, Kazemi M (2013) A genetic algorithm for solving a multi-floor layout design model of a cellular manufacturing system with alternative process routings and flexible configuration. Int J Adv Manuf Technol 66:845–865
29. Vin E, Delchambre A (2014) Generalized cell formation: iterative versus simultaneous resolution with grouping genetic algorithm. J Intell Manuf 25:1113–1124
30. Saeidi S, Solimanpur M, Mahdavi I, Javadian N (2014) A multi-objective genetic algorithm for solving cell formation problem using a fuzzy goal programming approach. Int J Adv Manuf Technol 70:1635–1652
31. Deep K, Singh PK (2015) Design of robust cellular manufacturing system for dynamic part population considering multiple processing routes using genetic algorithm. J Manuf Syst 35:155–163
32. Bhide P, Bhandwale A, Kesavadas T (2005) Cell formation using multiple process plans. J Intell Manuf 16:53–65
33. Alhourani F (2016) Cellular manufacturing system design considering machines reliability and parts alternative process routings. Int J Prod Res 54:846–863
34. De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6:239–251
35. Karoum B, Elbenani B, El Imrani AA (2016) Clonal selection algorithm for the cell formation problem. In: El Oualkadi A, Choubani F, El Moussati A (eds) Proceedings of the mediterranean conference on information and communication technologies 2015. Lecture notes in electrical engineering, vol 380. Springer, Cham, pp 319–326
36. Karoum B, El Khattabi N, Elbenani B, El Imrani AA (2016) An efficient artificial immune system algorithm for the cell formation problem. J Comput Methods Sci Eng 16:733744
37. Karoum B, Elbenani B (2017) A hybrid clonal algorithm for the cell formation problem with variant number of cells. Prod Eng Res Devel 11:19–28. doi:10.1007/s11740-016-0706-3
38. Moon YB, Chi SC (1992) Generalized part family formation using neural network techniques. J Manuf Syst 11:149–159
39. Sankaran S, Kasilingam RG (1990) An integrated approach to cell formation and part routing in group technology manufacturing systems. Eng Optim 16:235–245
40. Lee MK, Luong H, Abhary K (1997) A genetic algorithm based cell design considering alternative routing. Comput Integr Manuf Syst 10:93–108
41. Ouk Kim C, Baek JG, Baek JK (2004) A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences. Int J Prod Res 42:3911–3927
42. Alhourani F (2013) Clustering algorithm for solving group technology problem with multiple process routings. Comput Ind Eng 66:781–790
43. Sofianopoulou S (1999) Manufacturing cells design with alternative process plans and/or replicate machines. Int J Prod Res 37:707–720
44. Nagi R, Harhalakis G, Proth JM (1990) Multiple routings and capacity consideration in group technology applications. Int J Prod Res 28:1243–1257
45. Won Y, Kim S (1997) Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings. Comput Ind Eng 32:207–220
46. Hu L, Yasuda K (2006) Minimising material handling cost in cell formation with alternative processing routes by grouping genetic algorithm. Int J Prod Res 44:2133–2167