

A MULTI-PERIOD GROUP TECHNOLOGY METHOD FOR DYNAMIC CELLULAR MANUFACTURING SYSTEMS

J. Padayachee^{1*} & G. Bright¹

ARTICLE INFO

Article details

Submitted by authors 5 Apr 2016
Accepted for publication 28 Sep 2016
Available online 6 Dec 2016

Contact details

* Corresponding author
padayacheej@ukzn.ac.za

Author affiliations

¹ Discipline of Mechanical Engineering, University of KwaZulu-Natal, Durban, South Africa

DOI

<http://dx.doi.org/10.7166/27-4-1524>

ABSTRACT

A dynamic cellular manufacturing system is a system where cell configurations change in order to respond to changing production requirements. This paper proposes methods for multi-period cell and part family formation. The method of fuzzy C-means clustering is used for part family formation while balancing the part load between cells in each period; a genetic algorithm is presented to solve the model. An integer programming model is presented for multi-period cell formation with the objective of determining an optimal trade-off between reconfiguration cost and the cost of deficit capacity.

OPSOMMING

'n Dinamies sellulêre vervaardigingstelsel is 'n stelsel waarbinne selkonfigurasies verander in reaksie op veranderende produksie vereistes. 'n Metode van multi-periode sel en onderdeel familie formasie word voorgestel. Die wasige C-gemiddelde groepeeringsmetode word gebruik vir onderdeel familie vorming terwyl die onderdeel vereiste tussen die selle in elke periode gebalanseer word. 'n Genetiese algoritme word gebruik om die model op te los. 'n Heelgetal programmeringsmodel word voorgehou vir die multi-periode selfformasie om die optimale kompromie tussen herkonfigurasie koste en die koste van tekorte in kapasiteit te bereik.

1 INTRODUCTION

Cellular manufacturing systems (CMS) is a manufacturing system design paradigm that emerged in the 1970s and gained popularity in the 1980s [1]. The benefits of CMS include reduced part set-up times on machines, better organisation of tooling, lower work-in-process levels, reduced expenditure on material handling, less chaotic work flow, improved product quality, lower complexity scheduling, and hassle-free system management [1, 2].

High product variety, technology innovation, and unstable demand cause manufacturing systems to operate under rapidly-changing conditions [3]. Challenges include the frequent adaptation of manufacturing processes to new products, fluctuations in product demand and mix, and the manufacturing of multiple product types on the same equipment. The disadvantage of CMS is the lack of flexibility to accommodate changes to products and part mix, the under-use of machinery, and the cost incurred by having to re-layout cells [4]. A CMS will remain optimal over a planning horizon in which part variety and demand do not change [5]; however, the advantage of grouping a set of machines in one period may prove to be a disadvantage in the next period.

The concept of dynamic cellular manufacturing systems (DCMS) was proposed by Rheault, Drolet and Abdunour [6]. A DCMS is a system where the relocation of machines between cells is permitted, to create new configurations. DCMS can therefore be classified under the broader scope of reconfigurable manufacturing systems (RMS). In an RMS, changes in system structure are permitted and factored into the initial plant design in order to cope with future changes to products and markets [7].

The contribution of this research is to the use of multi-period group technology (GT) methods for identifying cells and part families. The method of fuzzy C-means clustering enables part family formation that balances the part load between cells in each period. An integer programming model is presented for multi-period cell formation with the objective of determining an optimal trade-off between reconfiguration cost and the cost of deficit capacity. Section 2 presents a review of recent DCMS literature, Section 3 presents the fuzzy C-means model and a genetic algorithm for its solution, and Section 4 presents a multi-period cell formation model. Section 5 demonstrates the use of individual methods to form part families and cells, and the paper concludes in Section 6.

2 LITERATURE SURVEY

CMS exploits the concept of group technology (GT) to cluster parts into families and machines into cells. The formation of machine cells and part families is a clustering problem that generally makes use of machine-component incidence data. A variety of GT methods have been developed to produce cells and families. Well-known methods for single-period cell and part family formation include: hierarchical similarity coefficient-based clustering by authors such as Seifoddini and Djassemi [8]; non-hierarchical clustering methods by Srinivasan and Narendran [9]; and matrix manipulation techniques, such as the modified rank order clustering (MODROC) developed by Chandrasekharan and Rajagopalan [10]. Recently, metaheuristics and artificial intelligence methods have been applied to the single-period GT problem; works include the application of a fuzzy adaptive resonance theory (ART) neural network by Won and Currie [11], and a bacteria foraging optimisation (BFO) algorithm by Nouri and Hong [12].

The concept of dynamic cellular manufacturing systems (DCMS) was proposed by Rheault *et al.* [6]. DCMS addresses multi-period cell formation under changing production requirements. The authors presented a cost-minimisation quadratic assignment model to aid with the decisions of relocating workstations and machinery between cells to achieve new system configurations. The method proposed did not include any long-term planning for re-layout, but rather served as a tool to determine the next configuration of the DCMS when the present configuration is no longer suitable.

Safaei, Saidi-Mehrabad, Tavakkoli-Moghaddam and Sassani [13] developed a mixed integer programme for a DCMS. The objective function minimises machine relocation, material handling, and reconfiguration costs. The formulation considers inter- and intra-cell material handling, alternative process plans, machine duplication, and re-layout via reconfiguration. A fuzzy programming-based approach was adopted to solve the model for a two-period, two-cell, eight-part, six-machine problem. In a similar work, Safaei, Saidi-Mehrabad and Jabal-Ameli [14] applied a hybrid simulated annealing algorithm to solve a larger three-period, two-cell, eight-part, six-machine problem. Safaei and Tavakkoli-Moghaddam [15] developed another similar Mixed Integer Programs (MIP) for a DCMS that considers the additional factors of subcontracting and inventory carrying costs. Mahdavi, Aalaei, Paydar and Solimanpur [16] proposed an integer programming model that considered worker assignment. Ossama, Youssef and Shalaby [17] considered the multi-period cell formation problem with the use of reconfigurable machines. The significance of reconfigurable machines in enabling new manufacturing system configurations has been highlighted by authors such as Landers, Min and Koren [18]; however, these machines have yet to gain significant industrial acceptance. Bayram and Sahin [19] developed a non-linear mixed integer programming model that considered both inter- and intra-cell material handling costs, costs for installing and uninstalling machines, and the possibility of purchasing new machines to determine new system configurations.

This research addresses a deficiency in previous DCMS methods where little or no consideration was given to equalising the distribution of the part load among individual cells. Cells that are highly loaded are difficult to manage, while cells with smaller part loads will exhibit low machine use. In addition to equalising part loads among cells, the reconfiguration of cells focuses on minimising any capacity, considering both part volumes and processing times. The multi-period machine assignment model presented here is also able to assign multiple machines.

3 THE FUZZY C-MEANS MODEL

The fuzzy C-means model is proposed as a way of identifying part families for DCMS. The fuzzy C-means (FCM) clustering model was first formulated by Dunn [20] and improved by Bezdek [21]. The

model is based on the concept of clustering around prototypes. Prototypes may be real elements from the data set, or artificial elements that act as the nucleus for cluster formation. The FCM model, according to Bezdek [21], is as follows:

$$\min \sum_{i \in V} \sum_{k \in Z} (\mu_{ik})^m D_{ik} \quad (1)$$

subject to:

$$\sum_{i=1}^C \mu_{ik} = 1, 1 \leq k \leq N \quad (2)$$

$$0 < \sum_{k=1}^N \mu_{ik} < N, 1 \leq i \leq C \quad (3)$$

$$\mu_{ik} \in [0,1], 1 \leq i \leq C, 1 \leq k \leq N \quad (4)$$

The objective function of the model minimises the product of membership values and the distance/dissimilarity between a data point k and the cluster prototype i , $k \in Z$, and $i \in V$. Here $Z = \{1, \dots, N\}$ is the set of elements to be clustered, and $V = \{1, \dots, C\}$ is the set of cluster prototypes. A cluster prototype is a data object that is representative of other elements of the cluster. The value m determines the fuzziness of clusters for $m > 1$; in this study $m = 2$ was used. The first constraint of the model ensures that the sum of memberships for each data element across all clusters equals one. The membership value is therefore comparable to a percentage. The second constraint ensures that all memberships are not allocated to a single cluster. The final constraint ensures that membership values are in the range $[0, 1]$.

3.1 Application to part family formation

The fuzzy C-means model can be applied to either part family formation or cell formation; the model is not able to cluster both machines and parts simultaneously. The output of the model is a set of cluster prototypes V and a membership matrix $U = [\mu_{ik}]_{C \times N}$. The model was chosen for part family formation, as the membership matrix can be used for multi-period family formation. The membership values indicate whether parts would be suitable members of more than one family. If a part has a high membership value in more than one family, the part may be distributed to the family that has fewer members, thereby facilitating a more even load balance among cells.

$$\mu_{ik} = \frac{\left(\frac{1}{D_{ik}}\right)^{1/(m-1)}}{\sum_{j=1}^C \left(\frac{1}{D_{jk}}\right)^{1/(m-1)}} \quad (5)$$

$$D_{JACCARD} = 1 - \frac{a}{a + b + c} \quad (6)$$

In order for the fuzzy C-means model to be applied to part family creation, a suitable measure of dissimilarity must be chosen and an appropriate membership model must be selected to calculate values of μ_{ik} . In this work, the original fuzzy C-means membership model, obtained from Bezdek [21] and shown in equation 5, was found to be suitable. The Jaccard distance measure, obtained from Hamers, Hemeryck, Herweyers, Janssen, Keters, Rousseau and Vanhoutte [23] and shown in equation 6, was chosen as the measure of dissimilarity among parts. The notation a is the total number of machines/processes required by both parts being compared, b is the number of machines/processes required exclusively by the first part, and c is the number of machines/processes required exclusively by the second part.

3.2 Solution by genetic algorithm

The fuzzy C-means model can be solved by a variety of methods, but most commonly by the fuzzy C-means (FCM) algorithm. The FCM algorithm cannot be applied to GT problems where part families are formed using data in a zero-one machine-component incidence matrix. The FCM algorithm uses Piccard iteration and involves a computation of new cluster prototypes that cannot be applied to zero-one data. The fuzzy C-means model for part families was therefore solved using a genetic algorithm (GA).

The purpose of the GA is to find the best set of cluster prototypes, thereby minimising the distance between data elements and prototypes. This consequently minimises the objective function of equation 1. A genetic algorithm consists of the following procedures: 1) initialisation, 2) evaluation, 3) selection, 4) crossover, 5) mutation, and 6) termination criteria check. Procedures 2-6 are repeated until the termination criteria are satisfied.

The initialisation procedure involves creating a population of solutions. The chromosome used to represent members of the population in the GA was a set of zero-one strings; the length of each string corresponded with the number of machine types available. A '1' indicates that the machine is used by the part, and a '0' indicates that the machine is not used. Consider the chromosome 'X':

$$X = \{x_1, x_2\}$$

Where:

$$\begin{aligned} x_1 &= [1, 1, 0, 0, 1, 1, 1, 1, 0] \\ x_2 &= [0, 1, 1, 1, 1, 0, 0, 0, 1] \end{aligned}$$

The zero-one strings, x_1 and x_2 , are substructures within the chromosome. Each substructure corresponds with one cluster prototype; for example, x_1 is the prototype of part family 1 and x_2 is the prototype of part family 2. The number of prototypes and substructures depends on the number of part families required by the user. The population was initialised using Bernoulli trials to generate random zero-one strings of the required length. The GA was found to perform better when actual part data was used to create members of the initial population; the part data stored in the columns of a machine-component incidence matrix were introduced as zero-one strings into the initial population (see Figure 1). In the GA used in this research, 50 per cent of the individuals in the initial population were created using a Bernoulli distribution to generate random chromosomes, and 50 per cent of the individuals were created by randomly selecting parts to serve as cluster prototypes.

The fitness evaluation for individuals of the population is performed by computing the Jaccard distance (equation 6) between each element in the set Z and each cluster prototype in the set V – where Z is the set of all parts and V is the chromosome of an individual (effectively $V=X$). Once the Jaccard distance is computed, the membership of each element in Z with respect to each cluster is calculated using equation 5. The objective function value is then computed using equation 1. Note that the GA does not have any explicit mechanism of ensuring that constraints (equations 2-4) are satisfied; however, the fuzzy C-means membership model (equation 5) ensures that membership values are in the range $\mu_{ik} \in [0, 1]$ and that the sum of memberships for an element across all clusters equals one. The membership model therefore satisfies constraint equation 2 and equation 4. Constraint equation 3 is rarely violated in any solution unless the number of natural clusters in the data set is actually one.

The strategy used to select parents of the next generation was stochastic uniform selection. The next generation of solutions was created by performing crossover and mutation operations on the selected parents. The crossover operation is performed by combining zero-one strings among parents; for example, if parent $X = \{x_1, x_2, x_3\}$ and parent $Y = \{y_1, y_2, y_3\}$, then two potential offspring may be $XY = \{x_1, x_2, y_2\}$ and $YX = \{y_1, x_3, y_3\}$. The number of strings to be included in offspring XY from parent X was determined by uniformly and randomly selecting an integer n in the interval $[1, C-1]$; recall that C is the specified number of clusters/prototypes. To complete offspring XY , $C-n$ strings are filled in from parent Y until the chromosome has a total of C strings. The remaining strings from parents X and Y , not used in offspring XY , are assigned to offspring YX . The specific strings to be assigned to offspring from parents are selected uniformly and randomly without replacement. Note that the order in which the strings appear in each set is of no significance to the objective function.

The bit flipping mutation mechanism was used in the GA. The number of strings in the chromosome selected for mutation was determined by uniformly and randomly selecting an integer n in the interval $[1, C]$. The specific strings to be mutated were selected uniformly and randomly without replacement. Bernoulli trials were used to determine whether a bit should be flipped; a single trial was applied to each bit with probability p . Here p is effectively the mutation rate, and a successful Bernoulli trial implies that the bit will be flipped. The GA terminated when the average relative change in fitness values over a specified number of generations was less than or equal to a specified

tolerance; the tolerance used was 1e-6. The GA also terminated when the maximum number of generations was reached.

4 THE FUZZY C-MEANS MODEL

In this section, a multi-period machine assignment model is presented. The objective is to find an optimal trade-off between reconfiguration cost and the cost of deficit capacity in cells. Reconfiguration cost refers to the cost of installing/uninstalling a machine in a cell. The choice to treat machine capacity in terms of hours also allows the model to allocate multiple machines of the same type to cells. The following notation is applicable to the model:

Sets:

$I = \{1, \dots, M\}$ is the set of machine types; $i \in I$.

$J = \{1, \dots, N\}$ is the set of part families; $j \in J$.

$K = \{1, \dots, T\}$ is the set of periods; $k \in K$.

Decision variables:

x_{ijk} is a integer variable that signifies the number of machines of type i allocated to the cell corresponding to part family j in period k .

u_{ijk} is an integer variable indicating hours of excess machine capacity with respect to machine type i , assigned to part family j in period k .

v_{ijk} is an integer variable indicating hours of deficit capacity with respect to machine type i and family j in period k .

p_{ijk} is an integer variable indicating the number of machines of type i added to the cell corresponding to family j in period k .

q_{ijk} is an integer variable indicating the number of machines of type i removed from the cell corresponding to family j in period k .

y_{ijk} is a zero-one variable that ensures that either u_{ijk} or v_{ijk} will assume a positive value while the other remains at zero.

Parameters:

H_{ijk} is the hours required on machine type i by parts belonging to family j in period k .

C_{ijk} is the cost of an hour of deficit capacity with respect to machine i and family j in period k .

W_k is the number of working hours in period k .

r_{ik} is the cost of a cell gaining/losing a copy of machine i in period k .

N_{ik} is the maximum number of machines of type i available for assignment in period k .

The model is as follows:

$$\min \text{Cost} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} r_{ik} (p_{ijk} + q_{ijk}) + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} C_{ijk} v_{ijk} \quad (7)$$

Subject to:

$$-u_{ijk} + v_{ijk} + W_k x_{ijk} = H_{ijk}, \forall i \in I, j \in J, k \in K \quad (8)$$

$$u_{ijk} \leq M y_{ijk}, \forall i \in I, j \in J, k \in K \quad (9)$$

$$v_{ijk} \leq M (1 - y_{ijk}), \forall i \in I, j \in J, k \in K \quad (10)$$

$$p_{ijk} - q_{ijk} - x_{ijk} + x_{ij(k-1)} = 0, \forall i \in I, j \in J, k \in K \quad (11)$$

$$\sum_{j \in J} x_{ijk} = N_{ik}, \forall i \in I, k \in K \quad (12)$$

$$y_{ijk} \in \{0,1\}; x_{ijk}, p_{ijk}, q_{ijk}, u_{ijk}, v_{ijk} \in Z^+ \quad (13)$$

The objective function minimises the sum of the reconfiguration cost and the sum of the cost of deficit capacity in cells. The first constraint determines the hours of surplus or deficit capacity with respect to the assignment of machine type i to part family j in period k . The second and third constraints are conditional constraints that ensure that either a capacity surplus or a deficit is reflected. A capacity surplus or deficit cannot be reflected simultaneously for the same 'ijk' combination. The zero-one variable y_{ijk} is used in conjunction with a constant M to implement the conditional constraints. M is an arbitrarily large constant; the only requirement on the constant M

is that it be larger than the largest possible capacity surplus or deficit. The fourth constraint determines whether machines of type i have been added or removed from the cell corresponding to family j in period k . The fifth constraint allows a maximum of N_{ik} machines of type i to be assigned to cells in time period k ; this ensures that the model does not assign more machines to cells than are available. The final constraint restricts all variables to positive integer values, with the exception of y_{ijk} , which is restricted to zero or one.

5 DCMS STUDY

5.1 Manufacturing requirements

A sub-contractor manufacturing company wishes to implement a DCMS that will fulfil manufacturing requirements over six years (i.e., six periods). During the period of six years, the manufacturer is contracted to produce 20 different parts; the machines and processing times are listed in Table 1, along with the part quantities. For this application, two cells/families were selected for implementation. In practice, the appropriate number of cells can be determined by creating a silhouette of the clusters; however, this is not discussed any further in this paper. (For further reading, see Rousseeuw [22].) The cost of an inter-cell movement per part is c_k , and the cost of a cell gaining or losing a machine is r_k ; these costs are listed in Table 2. At the point of designing a DCMS, the cost of equipment, such as dollies, used for inter-cell material handling can be integrated into the cost per part; another cost that could be added to c_k is the cost of hiring labour to perform the task. The cost r_k included the cost of installing/uninstalling machinery and of hiring technicians and machine-moving equipment. Since this is a relatively small manufacturing facility, the costs are not based on distances between cells. Note that here, r_k is a cost that applies to all machines; if each machine i has a different cost, then it is modelled as r_{ik} – as in the model in Section 4.

The purpose of reconfiguring cells is to eliminate inter-cell part flow, thereby reducing the complexity of scheduling and material handling. Reduced inter-cell part flow can also reduce work-in-process and shorten manufacturing lead times. Although the costs involved may be less significant than the benefits derived from new system configurations, having a cost trade-off as the objective can eliminate configuration changes to cells that are not very beneficial.

Table 1: Route sheet with part demand (Y=Year)

Part number	Sequence of machines	Process cycle time (minutes/operation)	Part volumes (thousands)					
			Y1	Y2	Y3	Y4	Y5	Y6
1	7-8-11-12	7-9-10-10	4.0	3.5	2.0	0.0	0.0	0.0
2	5-1-6-7-9	7-7-7-8-9	4.0	3.5	2.0	0.0	0.0	0.0
3	6-7-8-9	7-7-7-9	0.0	6.0	9.0	6.4	3.0	0.0
4	1-4-6-9	7-10-7-7	8.0	7.0	4.0	0.0	0.0	0.0
5	10-1-7-9-12	10-7-8-8-8	0.0	0.0	0.0	1.5	3.0	5.0
6	5-1-4-6-9	8-9-9-8-9	4.0	3.5	2.0	0.0	0.0	0.0
7	2-8-12-10	10-8-9-10	4.0	3.5	2.0	0.0	0.0	0.0
8	2-6-8-5	8-7-10-8	0.0	0.0	0.0	3.0	6.0	10.0
9	1-4-12-5	9-8-7-7	0.0	6.0	9.0	6.4	3.0	0.0
10	2-3-8-5	9-8-7-10	0.0	3.0	4.5	4.7	4.5	5.0
11	2-8-11-12-10	9-10-8-7-10	4.0	3.5	2.0	0.0	0.0	0.0
12	2-3-4-6-8	10-9-10-7-9	0.0	0.0	0.0	1.5	3.0	5.0
13	1-2-3-4-12	9-8-9-10-9	0.0	3.0	4.5	3.2	1.5	0.0
14	2-8-12-10	10-8-9-9	8.0	7.0	4.0	0.0	0.0	0.0
15	10-1-7-9-12	9-8-8-9-8	0.0	0.0	0.0	1.5	3.0	5.0
16	5-1-3-6-7	7-9-10-8-7	4.0	3.5	2.0	0.0	0.0	0.0
17	1-7-11-12	8-7-9-10	0.0	0.0	0.0	3.0	6.0	10.0
18	2-3-4-12	10-10-9-9	0.0	3.0	4.5	4.7	4.5	5.0
19	6-7-8-9-11	8-7-9-8-9	0.0	3.0	4.5	3.2	1.5	0.0
20	10-2-6-8-9-11	9-9-7-7-8-10	0.0	3.0	4.5	3.2	1.5	0.0

Table 2: Cost of inter-cell machine and part relocation

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
r_k	R 4000.00	R 4200.00	R 4400.00	R 4600.00	R 480.00	R 500.00
c_k	R 1.50	R 1.60	R 1.70	R 1.80	R 1.90	R 2.00

5.2 Manufacturing requirements

In order to form part families, the relationship between parts and machines is first represented as a machine-component incidence matrix. The matrix derived from the machine part relationships, shown in Table 1, is now presented in Figure 1. The data in Figure 1 allows each part to be represented as a data element Z_n before fuzzy clustering is applied; for example, the data element corresponding to part 1 is $Z_1 = [0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1]$. The GA was used for part family formation with a mutation rate of 0.1 and a population size of 40, and was set to terminate if, after 100 generations, no improvement was found. The algorithm terminated after 126 generations with a final objective function value of 7.42081, as shown in Figure 2.

		Parts																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Machines	1	0	1	0	1	1	1	0	0	1	0	0	0	1	0	1	1	1	0	0	0
	2	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	1	0	1
	3	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	0
	4	0	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	0
	5	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0
	6	0	1	1	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	1	1
	7	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0
	8	1	0	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1
	9	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1
	10	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	0	0	0	0	1
	11	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1
	12	1	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	1	0	0

Figure 1: Machine-component incidence matrix

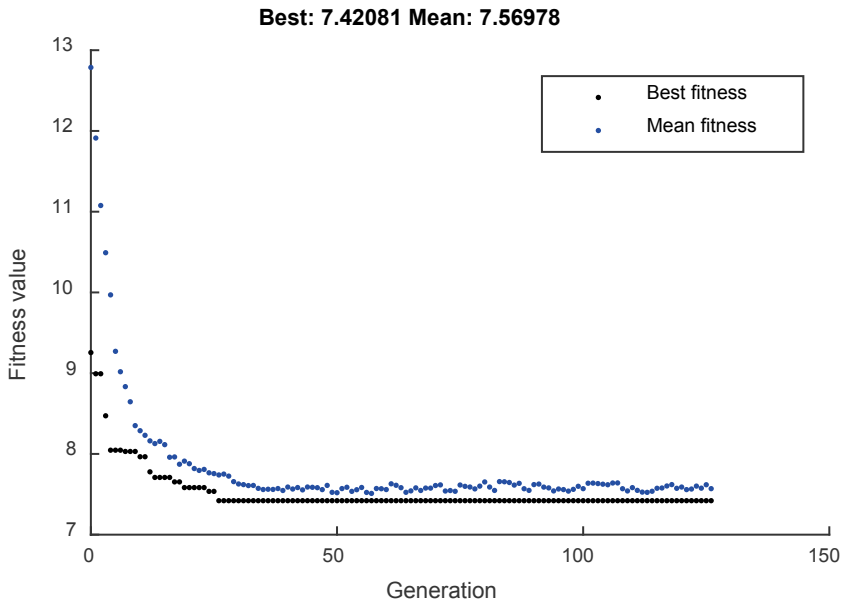


Figure 2: Genetic algorithm convergence

The application of the fuzzy C-means model with the GA solver produced the following membership matrix:

$$U = \begin{bmatrix} 57 & 45 & 52 & 46 & 59 & 41 & 57 & 46 & 42 & 42 & 63 & 41 & 41 & 57 & 59 & 41 & 57 & 42 & 59 & 67 \\ 43 & 55 & 48 & 54 & 41 & 59 & 43 & 54 & 58 & 58 & 37 & 59 & 59 & 43 & 41 & 59 & 43 & 58 & 41 & 33 \end{bmatrix} \times 10^{-2}$$

The membership matrix $U=[\mu_{ik}]$ consists of two rows: the first row corresponds with family 1 and the second with family 2. The membership values are presented as a percentage; for example, part 1 in column 1 has 57 per cent membership in family 1 and 43 per cent membership in family 2. Parts may be allocated to families according to maximum membership or combined with other criteria. In this study, parts with membership values in the range of 45-55 per cent were moved between part families at different times in order to balance the part load in corresponding cells. Parts 2, 3, 4, and 8 have membership values in the range 45-55 per cent, implying that these parts are on the ‘border’ of the two part family clusters. Table 3 shows the assignment of ‘border parts’ to part families in order to balance the part load in each cell.

Table 3: Assignment of border parts (P=Part)

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Initial load - Cell 1	20000	23500	19000	12400	15000	20000
Initial load - Cell 2	8000	22000	26500	20500	16500	15000
Demand - P2	4000	3500	2000	0	0	0
Demand - P3	0	6000	9000	6400	3000	0
Demand - P4	8000	7000	4000	0	0	0
Demand - P8	0	0	0	3000	6000	10000
Family 1		P4	P2, P3	P3, P8	P8	
Family 2	P2, P4	P2, P3	P4		P3	P8
Final load - Cell 1	20000	30500	30000	21800	21000	20000
Final load - Cell 2	20000	31500	30500	20500	19500	25000

To develop this table, parts with membership values outside of the range 45-55 per cent are automatically assigned to either family 1 or family 2, based on maximum membership; the demand volumes are then attributed to the corresponding cell to determine initial part loads. The initial part families are:

- $family\ 1 = \{P1, P5, P7, P11, P14, P15, P17, P19, P20\}$,
- $family\ 2 = \{P6, P9, P10, P12, P13, P16, P18\}$.

Parts in the initial families are not swapped between families to balance the part load in each cell. Border parts and demands are then listed in the table and these parts are assigned to either family 1 or family 2, across multiple periods, to equalise the corresponding part load in each cell. The procedure for assigning border parts is to assign the parts with the highest demands first to equalise the load on each cell. In this study, the sum of all load imbalances between cell 1 and cell 2 across all six periods was 9300 parts. If this approach had not been used, and every part had been allocated to the cell of maximum membership, the sum of all load imbalances across the six periods would have been 21700 parts.

5.3 Multi-period cell formation

5.3.1 Parameter calculations

Multi-period cell formation was performed by solving the model presented in Section 5; the set-up of the model required that parameters H_{ijk} , C_{ijk} , W_k , r_{ik} , and N_{ik} be determined. The cost of reconfiguring r_{ik} was presented in Table 2; here, the assumption was made that the cost of reconfiguration is the same for all machines i , $i \in I$. The hours of machine capacity H_{ijk} , required of machine i with respect to part family j in period k , is determined by adding the product of part volume and processing times of each part p belonging to family j and dividing by 60, as shown here:

$$H_{ijk} = \frac{\sum_{p=1}^P volume_{pjk} \times minutes_{ip}}{60} \quad (14)$$

For this study, each period consisted of $W_k = 5300$ working hours, and machine availability was taken as 80 per cent. The number of machines required by the system was then calculated by equation 15, where $H_{ik} = H_{i1k} + H_{i2k}$:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} \quad (15)$$

The application of equation 15 determined that one copy of each machine was required to meet the capacity requirements of the factory across all six periods, with the exception of machine type 8, where two copies of the machine were required in the second year. The final parameter that required estimation was C_{ijk} , which is the cost of an hour of deficit capacity with respect to machine i and the cell corresponding to part family j in period k . The method of estimating this coefficient is based on the assumption that parts will travel to other cells to access the required capacity that is at a deficit in a resident cell. Equation 16 estimates the number of parts affected by a deficit hour of capacity by dividing 60 minutes by the weighted average processing time \bar{t}_{ij} , with respect to parts in family j . The number of parts multiplied by the cost of a single inter-cell part movement c_k gives the coefficient C_{ijk} :

$$C_{ijk} = \left\lceil \frac{60}{\bar{t}_{ij}} \right\rceil c_k \quad (16)$$

5.3.2 Model solution

Table 4: Six period assignment of machines to cells

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

The cell formation model was solved by the MathWorks® *intlinprog* solver, which is a mixed-integer linear programming solver. The solver implements a branch and cut algorithm that was able to determine the best solution by cut generation and pre-processing at the outset, without having to apply branching procedures. The solver applied 273 implication cuts and one strong Chvátal-Gomory cut. Table 4 above presents the assignment of machine types 1-12 to cells in each period. Blocks shaded in black indicate when a cell gains a machine, and blocks shaded in grey indicate when a cell loses a machine. The total cost of the solution is R 454 860.00. This cost includes the cost of the initial placement of machines.

Table 5 shows the total hours of deficit capacity in cells in each period. This information is derived directly from the final values of variable v_{ijk} in the assignment model. Since enough machines were made available to the system, according to equation 15, this deficit capacity in cells is an indicator of parts having to perform an inter-cell material handling movement to access the required machine capacity in another cell. This information is also useful, as it indicates the extent to which machines will spend time processing parts that are from another cell. Consider, for example, machine 9 in cell 2 and period 2: 1950 hours of machining are required on the machine; however, the machine is

located in cell 1 in that period. Machine 12 in cell 1 will therefore spend 36.79 per cent ($1950/5300 = 0.3679$) of its operating time processing parts from another cell.

Table 5: Hours of deficit capacity in cells (M = Machine)

Cell 1												
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Year 1	0	0	0	0	0	0	467	0	0	0	0	0
Year 2	817	0	0	1167	0	1567	759	0	0	0	0	0
Year 3	234	1975	0	0	234	0	0	0	0	0	0	1467
Year 4	775	880	0	0	400	0	0	0	0	0	0	900
Year 5	0	1025	0	0	800	0	0	0	0	0	0	0
Year 6	0	0	0	0	0	0	0	0	0	0	0	0
Cell 2												
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Year 1	0	0	0	0	0	0	0	0	0	0	0	0
Year 2	0	1350	0	0	0	0	0	0	1950	0	0	1600
Year 3	0	0	0	0	0	1000	234	0	767	0	0	0
Year 4	0	0	0	0	0	175	0	0	0	0	0	0
Year 5	675	0	0	0	0	700	350	0	450	0	0	1250
Year 6	0	0	0	0	0	0	0	0	0	0	0	750

6 CONCLUSION

This paper has presented methods for multi-period machine cell and part family formation. The research addresses a deficiency in previous DCMS methods where little or no consideration was given to equalising the distribution of the part load among individual cells. The application of the methods was demonstrated on a 12-machine, 20-part problem across six periods of time with varying part demand. The approach of fuzzy clustering was used in part family formation. The membership matrix, which is an output of the fuzzy clustering process, was used to identify members of part families that would not change over the planning horizon; other 'border' parts with membership values in the range of 45-55 per cent were proposed for moving between part families to equalise the part load among cells as best as possible. An assignment model was developed for multi-period cell formation; in this model, machine capacity was treated in terms of operating hours available to a part family. Unlike traditional cell formation methods, the assignment model is able to allocate more than one machine of the same type to cells when there is a need for greater capacity. The cost coefficient for an hour of deficit capacity was based on the number of parts that would have to perform an inter-cell movement to access necessary capacity. The model focused on determining an optimal economic trade-off between relocating parts and relocating machines between cells. The method of fuzzy clustering is a versatile technique for both machine cell and part family formation. The possibility exists for multiple types of membership criteria to be applied to parts or machines; for example, additional membership matrices could be developed, based on geometric similarity, size similarity, and tooling similarity. The use of multiple membership matrices to create evolving part families is a topic for future research.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Technology Innovation Agency (TIA) – South Africa for supporting this research under the reconfigurable manufacturing systems grant.

REFERENCES

- [1] Hyer, N. and Wemmerlov, U. 2002. *Reorganizing the factory: Competing through cellular manufacturing*, 1st ed. United States of America: Productivity Press.
- [2] McLaughlin, P. and Durazo-Cardenas, I. 2013. Cellular manufacturing applications in MRO operations, *Procedia CIRP*, 11, pp. 254-259.
- [3] Tavakkoli-Moghaddam, R., Aryanezhad, M.B., Safaei, N. and Azaron, A. 2005. Solving a dynamic cell formation problem using metaheuristics, *Applied Mathematics and Computation*, 170(2), pp. 761-780.
- [4] Irani, S.A., Subramanian, S. and Allam, Y.S. 2007. Introduction to cellular manufacturing systems, in *Handbook of cellular manufacturing systems*, ed. John Wiley & Sons, Inc., pp. 1-23.
- [5] Shiyas, C.R. and Pillai, V.M. 2014. A mathematical programming model for manufacturing cell formation to develop multiple configurations, *Journal of Manufacturing Systems*, 33(1), pp. 149-158.
- [6] Rheault, M., Drolet, J.R. and Abdounour, G. 1996. Dynamic cellular manufacturing system (DCMS), *Computers & Industrial Engineering*, 31(1), pp. 143-146.
- [7] Koren, Y. and Shpitalni, M. 2010. Design of reconfigurable manufacturing systems, *Journal of Manufacturing Systems*, 29(3), pp. 130-141.
- [8] Seifoddini, H. and Djassemi, M. 1991 The production data-based similarity coefficient versus Jaccard's similarity coefficient, *Computers & Industrial Engineering*, 21(1), pp. 263-266.
- [9] Srinivasan, G. and Narendran, T.T. 1991. GRAFICS: A nonhierarchical clustering algorithm for group technology, *International Journal of Production Research*, 29(3), pp. 463-478.
- [10] Chandrasekharan, M.P. and Rajagopalan, R. 1986. MODROC: An extension of rank order clustering for group technology, *International Journal of Production Research*, 24(5), pp. 1221-1233.
- [11] Won, Y. and Currie, K.R. 2007. Fuzzy ART/RRR-RSS: A two-phase neural network algorithm for part-machine grouping in cellular manufacturing, *International Journal of Production Research*, 45(9), pp. 2073-2104.
- [12] Nouri, H. and Hong, T.S. 2013. Development of bacteria foraging optimization algorithm for cell formation in cellular manufacturing system considering cell load variations, *Journal of Manufacturing Systems*, 32(1), pp. 20-31.
- [13] Safaei, N., Saidi-Mehrabad, M., Tavakkoli-Moghaddam, R. and Sassani, F. 2008. A fuzzy programming approach for a cell formation problem with dynamic and uncertain conditions, *Fuzzy Sets and Systems*, 159(2), pp. 215-236.
- [14] Safaei, N., Saidi-Mehrabad, M. and Jabal-Ameli, M.S. 2008. A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system, *European Journal of Operational Research*, 185, pp. 563-592.
- [15] Safaei, N. and Tavakkoli-Moghaddam, R. 2009. Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems, *International Journal of Production Economics*, 120(2), pp. 301-314.
- [16] Mahdavi, I., Aalaei, A., Paydar, M.M. and Solimanpur, M. 2010. Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment, *Computers & Mathematics with Applications*, 60(4), pp. 1014-1025.
- [17] Ossama, M., Youssef, A.M.A. and Shalaby, M.A. 2014. A multi-period cell formation model for reconfigurable manufacturing systems, *Procedia CIRP*, 17(2014), pp. 130-135.
- [18] Landers, R., Min, B. and Koren, Y. 2001. Reconfigurable machine tools, *CIRP Annals – Manufacturing Technology*, 50 (1), pp. 269-274.
- [19] Bayram, H. and Şahin, R. 2016. A comprehensive mathematical model for dynamic cellular manufacturing system design and linear programming embedded hybrid solution techniques, *Computers & Industrial Engineering*, 91(2016), pp. 10-29.
- [20] Dunn, J.C. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics*, 3(3), pp. 32-57.
- [21] Bezdek, J. (ed.) 1981. Models for pattern recognition, in *Pattern recognition with fuzzy objective function algorithms*. US: Springer, pp. 1-13.
- [22] Rousseeuw, P.J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, 20(1987), pp. 53-65.
- [23] Hamers, L., Hemeryck, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R. and Vanhoutte, A. 1989. Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula, *Information Processing & Management*, 25(3), pp 315-318.