Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management
Istanbul, Turkey, July 3 – 6, 2012

Integrated Cellular Manufacturing System Design: an Evolutionary Algorithm Approach

Michael Mutingi
Department of Mechanical Engineering
University of Botswana, Gaborone, Botswana

Charles T. Mbohwa
Department of Quality and Operations Management
University of Johannesburg, Johannesburg, South Africa

Samson Mhlanga and William Goriwondo
Department of Industrial and Manufacturing Engineering
National University of Science and Technology, Bulawayo, Zimbabwe

Abstract

Cellular manufacturing system design has received much attention for the past three decades. The design process involves decisions on (i) cell formation, (ii) cell layout, and (iii) layout of cells on the shop floor. These decisions should be addressed jointly, if full benefits of cellular manufacturing are to be realised. However, due to the complexity of the problem, most researchers addressed these phases sequentially. In this paper, we propose an enhanced evolutionary algorithm to jointly address cell formation and layout problems, based on sequence data. The approach compares favourably to well-known heuristics and performed well on published data sets, providing improved solutions.

Keywords
Evolutionary algorithms, cellular manufacturing, cell formation, group layout

1. Introduction
Cellular Manufacturing System (CMS) is an application of the concepts of group technology aimed at improving productivity and flexibility in modern manufacturing systems (Mahdavi et al., 2010; Yin et al., 2005; Sarker and Xu, 1998). It requires decomposition of a manufacturing system into autonomous manufacturing cells so as to enhance tooling, material handling, scheduling and shop-floor control. Part families with similar processes or design features are grouped into machine cells so that each part family can possibly be processed entirely in a single cell. Machine layout within each cell is crucial in order to improve efficiency and effectiveness of the overall production system. Setup times, work-in-process inventories, and throughput times can be reduced significantly. CMS design and layout involves three main decisions: (i) *cell formation*: grouping of machines which can operate on a product family, (ii) intra-cell layout: layout machine within each cell, (iii) *inter-cell layout* - layout of cells with respect to one another, and (iv) *group scheduling*: scheduling of parts for production. Ideally, these decisions should be addressed simultaneously so as to obtain the best possible results (Mahdavi and Mahadevan, 2008; Kaebernick and Bazargan-Lari, 1996). However, due to the complexity of the decision problem and the limitations of conventional approaches, earlier studies focused on these decisions sequentially (Mahdavi et al., 2010; Yin and Yasuda, 2004; Onwubolu and Mutingi, 2001). Major objectives considered in assessing the quality of solutions include inter-cell and intra-cell movements, utilization and material handling costs.

Previous studies have relied on the use of zero-one machine-component incidence matrix as input data for the Cell Formation Problem (CFP). Sequence data, which shows job flow patterns, provides vital additional information for CMS design as well as machine layout within a cell. However, a few researchers have recently made efforts to use sequence data for cell layout (Yin et al., 2005; Won and Lee, 2001; Jayaswal and Adil, 2004). Nevertheless, the cell formation and layout problem are often treated independently. In an attempt to jointly address the cell formation and layout problem, various researchers and practitioners used a sequential approach where cells are formed first, followed by intra-cell layout construction. Since the final solution largely depends on the initial cell formation, the quality of the final solution is compromised. The basic CFP is *NP*-complete, meaning that it has no known polynomial time algorithm due to its combinatorial nature (Kumar et al., 1986). It follows that the integrated cell formation and layout problem is also *NP*-complete. Therefore, the use of heuristic approaches such as simulated annealing (Ji-Yang, 2010), tabu search (Wu et al., 2009) and evolutionary algorithms (Gan and Zheng, 2011) is quite appropriate. In this study, an enhanced Evolutionary Algorithm (EA) is proposed for integrated cell formation and layout design based on sequence data. The approach utilizes sequence data to identify machine cells as well as machine layout within each cell. In view of this, the major objectives for this paper are:

(1) To develop an EA for solving the integrated CMS design and layout problem using sequence data.
(2) To develop relevant performance metrics to address the integrated cell formation and layout problem.

The next section gives a description of the joint cell formation and layout problem. Section 3 provides a brief background to EA, followed by an outline of the proposed EA scheme for cell formation and layout in Section 4. Results and discussions are presented in Section 5. Section 6 concludes the paper.

2. Problem description: cell formation and layout

Cell formation in CMS involves grouping of machines which can operate on a product family with similar manufacturing processes and features such that little or no inter-cell movement of products is involved. On the other hand, cell layout problem involves layout machine within each cell and layout of cells with respect to one another. The overall objective of cell formation is to gain the advantages of group technology. In assessing the quality of solutions, common objectives considered are (i) minimization of inter-cell movements, (ii) minimization of intra-cell movements, (iii) maximization of utilization, (iv) minimization of material handling costs, and (v) minimization cell work-load imbalances. The joint cell formation and layout problem is a new approach that seeks to identify manufacturing cells and the layout (sequence) of machines in the cells in an integrated manner. This approach seeks to avoid compromising the quality of solutions with respect cell formation and cell layout objectives.

3. Evolutionary Algorithms

Evolutionary algorithms are search methods inspired by the philosophy of natural selection and survival of the fittest (Back, 1996). Unlike traditional optimization techniques, EA searches from a population of solutions, rather than from a single point. Iterations involves a competitive selection that weeds out poor solutions. Solutions with high fitness are recombined with other solutions by swapping parts of a solution with another. Additionally, solutions are mutated by making small changes to elements of the solutions. Recombination and mutation are used to generate new solutions that are biased towards visited regions of space with good solutions. EA methodologies have been applied extensively in combinatorial problems in engineering, telecommunications, sciences, agriculture, business, and manufacturing (Goldberg, 1989). The method integrates the elements of stochastic and direct search to obtain optimal (or near-optimal) solutions within reasonable computation time. The general EA scheme is shown in Figure 1.

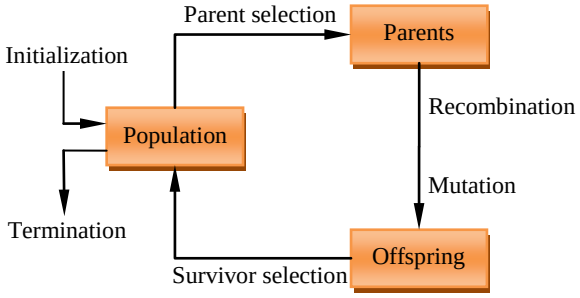

Figure 1: The general framework of EA as a flow chart

EAs offer unique advantages over other stochastic searches, including implicit parallelism, independence from gradient information, and flexibility to hybridization with other heuristics. Some early applications on the cell formation problem are found in (Venugopal and Narendran, 1992; Hsu and Su, 1998). Significant improvements to EAs have been made for specific problem areas. Realizing some shortcomings of classical genetic algorithms (GAs) for grouping problems, Falkenauer (1992) introduced group genetic algorithm (GGA) specifically designed to handle special structures of grouping/clustering problems. Some applications of GGA exist in literature (James et al., 2010; Filho and Tiberti, 2006). Based on GGA principles, an enhanced EA approach is proposed for the joint machine cell formation and layout problem.

4. EA approach for cell formation and layout

The proposed EA approach integrates genetic features inherent in grouping problems with the power of local search in order to refine new chromosomes generated. EA is a favourable approach compared to other heuristic and conventional approaches. The EA design for the joint cell design and layout problem is presented based on its six main elements:

(1) Objective function, or fitness function
(2) Chromosome encoding scheme.
(3) Generation of initial population.
(4) Selection and recombination strategies.
(5) Genetic operators: crossover, mutation and inversion.
(6) Diversification.

4.1 Objective/fitness function formulations

From a cell formation viewpoint, the presence of exceptional parts and voids should be minimized. In layout design, adjacency of machines in a cell is key as it can reduce material handling costs significantly (Grefenstette, 1987). From a production planning viewpoint, the sequence in which machines are placed in cells often creates unwanted reverse flows. Consider a typical solution for a simple cell formation problem in Table 1. A cell with machines 1 and 2 has two possible sequences (layouts), that is, (1, 2) and (2, 1). While cell layout (1, 2) has three consecutive forward flows due to part 1, 3 and 7, layout (2,1) has only two. Therefore, layout (1, 2) is preferred.

Table 1: A typical solution for a cell formation problem

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

An ideal objective function must capture and evaluate the effects of the sequence of machines within cells. One basic way of evaluating the fitness of a cell layout is to express the objective function in terms of number of consecutive forward flows of parts. In this connection, Mahdavi and Mahadevan (2008) proposed two performance indices: (i) cell flow index (CFI) and (ii) overall flow index (OFI). The following notation is used in this model;

n: number of parts in the system
m: number of machines in the system
n_c: number of parts in cell c
m_c: number of machines in cell c
v_c: number of voids in cell c
N_{fc}: number of consecutive forward flows within cell c
S_{jk}: machine-component matrix $[s_{jk}]$; $s_{jk} = 1$ if part k visits machine j, and 0 otherwise

To determine the cell flow and overall flow performance measures, the total number of operations and the consecutive flows between a pair of machines are calculated. The total number of flows N_{flow} is:

$$N_{flow} = \sum_{k} \max_{j} (s_{jk} - n) \tag{1}$$

Therefore, the total number of flows in each cell c is determined be the following expression;

$$N_{tc} = (n_c m_c) - v_c - n_c \tag{2}$$

In addition to the above, the cell flow index for cell c, CFI_c, is the ratio of the number of consecutive forward flows to the total number of flows within the cell;

$$\text{CFI}_c = \frac{N_{fc}}{N_{tc}} \tag{3}$$

The average cell flow index (ACFI) is the weighted average of the CFIs. This measure is given by F1as follows;

$$F1 = (1/n) \cdot \sum_{c} n_c \text{CFI}_c \tag{4}$$

The overall cell flow index (OFI) defines the ratio of the sum of consecutive forward flows in all the cells to the total number of the flows required to process all the parts. OFI is defined by F2 as;

$$F2 = \left(1/N_{nflow}\right) \cdot \sum_{c} N_{fc} \tag{5}$$

In this analysis, as the number of voids in the cell decreases and/or the number of consecutive forward flows increase, CFI increases. It follows that ACFI represents the solution quality with respect to the number of voids and the intra-cell moves. While ACFI points to the intra-cell movements, the OFI measures inter-cell movements. As such, a combination of these performance measures ensures that the cell formation and layout are addressed jointly. Let w_1 and w_2 denote the weights of *F1* and *F2*, respectively. Then, the weighted objective function is;

$$F = w_1 F_1 + w_2 F_2 \tag{6}$$

4.2 Chromosome representation

EA's performance strongly relies on the chromosome (string) coding scheme used. Most coding schemes in literature use strings of integer numbers where the position of the number represents the machine and the value of the number identifies the cell number. A typical chromosome [2 3 1 1 2 3 1 1], containing 8 machines represents a manufacturing system with 3 cells: machines 1, 5, 6 and 8 are in cell 1, machines 2 and 3 are in cell 2, and machines 4, 7, 9 occupy cell 3. The proposed EA algorithm uses an improved coding scheme that utilizes a group structure for each feasible string based on three code schemes (see Figure 2), similar to the one proposed by Filho and Tibert(2006). The first code scheme, code 1, is a string of size m, where m represents is the number of machines in

the system. The second is a group structure upon which the genetic operators act. The third denotes the last position of each group/cell.

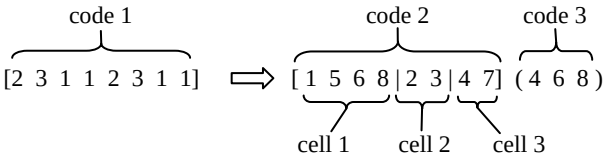

Figure 2: Chromosome representation

Code 3 shows that the first four machines (genes) are in cell 1, the next two genes are in cell 2, and the last two machines are in cell 3. In this EA implementation, several enhanced features are developed in of objective/fitness functions formulation, genetic operators, chromosome repair and other genetic strategies.

4.3 Initial population

An initial population of the desired size (*popsize*), is generated randomly. Consider a typical problem with *m* machines, where the desired number of cells is *v*. Assuming each cell comprises at least two machines, the initial population is created according to the following procedure:

Repeat

1. For each cell *j* (*j*=1,...,*v*), randomly select two machines from the set of machines.
2. For the remaining (*n*-2*j*) unassigned machines, randomly assign a machine to a cell.
3. Encode the chromosome using code 1 and add to the initial population.

Until (population size *popsize* is achieved).

In every EA application, the goal is to minimize some cost function which is usually mapped to a score function that evaluates the generated chromosomes. A mapping procedure initially suggested by Goldberg (1989) is applied;

$$f^{i}(t) = \begin{cases} f^{i}_{\max} - g^{i}(t) & \text{if } g_{i}(t) < f^{i}_{\max} \\ 0 & \text{if otherwise} \end{cases} \tag{7}$$

where, $g(t)$ is the objective function of a chromosome and $f_{\max}$ is the maximum objective function in the population.

4.4 Selection strategy

A number of selection strategies have been suggested by Goldberg (1989), including deterministic sampling, remainder stochastic sampling with/without replacement, stochastic tournament, and stochastic sampling with/without replacement. The remainder stochastic sampling without replacement has been found to be the most effective and is applied in this work. In this strategy, each chromosome *i* is selected and stored in the mating pool according to the expected count e_i calculated as,

$$e_{i} = \frac{f_{i}}{(1/popsize)\sum_{i=1}^{s} f_{i}} \tag{8}$$

Each chromosome receives copies equal to the integer part of e_i, while the fractional part is treated as success probability of obtaining an additional copy of each chromosome.

4.5 Genetic operators

This section outlines the design issues relating to genetic operators for the proposed EA for the cell design and layout problem are defined. Unique crossover, mutation and inversion strategies are developed.

4.5.1 Crossover

Crossover is a probabilistic evolutionary mechanism which seeks to mate selected chromosomes in order to produce a pool of new offspring, called selection pool (*spool*). It allows the algorithm to generate new solutions and to explore unvisited regions in the solution space. The proposed group crossover operator exchanges groups of genes of selected chromosomes (see Figure 3). Crossover operations occur with probability *pcross* until the desired pool size, $poolsize = popsize \cdot pcross$, is obtained. The procedure for the group crossover operator is:

Repeat

1. Generate a random integer number between 1 and (*v*-1), where *v* is the number of cells. The generated number defines the crossover point.
2. Swap the groups to the right of the crossover point to generate two offspring.
3. Repair the offspring by eliminating any duplicated machines and introducing missing machines.

Until (selection *poolsize* is achieved).

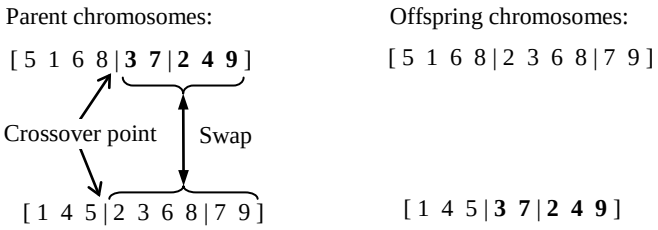

Figure 3: Crossover operator

After crossover, some machines may appear in more than one cell, and some may be missing. Such offspring should be repaired. The repair procedure identifies duplicated machines and eliminates those to the left of the crossover point. Missing machines are inserted into the cell with the least number of nodes (see Figure 4). Thus, the group representation scheme enhances the crossover operator by taking advantage of the group structure.

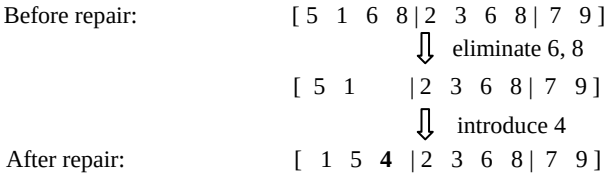

Figure 4: Chromosome repair procedure

4.5.2 Mutation

The mutation operator is applied to every new chromosome with probability *pmutation* in order to maintain diversity of the population and avoid premature convergence. Two mutation operators are proposed, namely *swap mutation* and *shift mutation*. Swap mutation operates by exchanging genes between two randomly chosen groups in a chromosome (see Figure 5.). Its general procedure can be summarized as follows:

1. Randomly select two integer numbers from the set {1, 2,..., *v*}, where *v* is the number of cells or groups.
2. Randomly choose a gene from each group.
3. Swap the selected genes, exchanging their values.

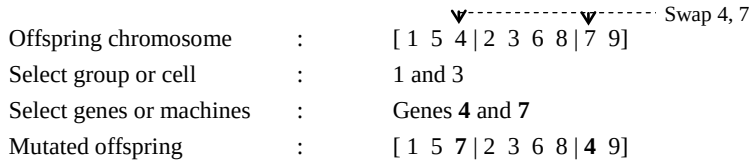

Figure 5: Swap mutation

The *shift mutation operator* works by shifting the frontier between two adjacent groups by one step either to the right or to the left, as shown in Figure 6. Essentially, the number of nodes is increased in one group and simultaneously decreased in the other. The procedure for the mutation operator is summarized thus;

1. Generate a random integer number between 1 and (*v*-1). Let this number represent the chosen frontier.
2. Randomly choose the direction of shift: *Right* or *Left*.
3. Shift the frontier in the selected direction, thereby moving one node between adjacent groups.

Offspring chromosome	:	[1 5 4 | 2 3 6 8 | 7 9]
Select frontier, rand (1, 2):		1 ↘ Shift frontier
Select direction	:	Right
Mutated offspring	:	[1 5 4 **2** | 3 6 8 | 7 9]

Figure 6: Shift mutation operator

4.5.3 Inversion operator

In order to curb premature convergence of the population, an inversion is applied, at a very low probability, on chromosomes selected for crossover operation. Basically, the inversion strategy operates by rearranging the groups in the reverse order, for instance, the order of cells (1, 2, 3) is transformed to cells (3, 2, 1) as illustrated below.

Chromosome before inversion	:	[1 5 6 8 | 3 7 | 2 4 9]
Chromosome after inversion	:	[2 4 9 | 3 7 | 1 5 6 8]

4.5.4 Diversification and convergence

As iterations proceed, the population converges to a particular solution. Rapid loss of diversity and premature convergence may occur before an optimal solution is obtained. To track the diversity of the solution space, Grefenstette (1987) proposed an entropic measure H_i in a population of candidates. For each machine *i*, H_i is defined;

$$H_i = \sum_{j=1}^{m} \frac{(n_{ij}/p) \cdot \log(n_{ij}/p)}{\log(m)} \tag{9}$$

Here, n_{ij} is the number of strings in which machine *i* is assigned position *j* in the current population, *p* is the solution space size, and *m* is the number of machines. Divergence *H* is then calculated as;

$$H = \sum_{i=1}^{m} H_i / m \tag{10}$$

As iterations advance, the divergence parameter *H* approaches zero. Therefore, the diversity of the solution space can be monitored and controlled by applying the inversion operator to improve diversity to a preset value. In order to prevent loss of good solutions, a fraction (e.g., 0.2) of best performing solutions from the undiversified population is preserved. Candidate solutions from the diversified population are compared with those from the diversified population. The best performing candidates are taken into the next generation.

4.6 The EA implementation

The structure of the proposed EA for solving the integrated cellular manufacturing system problem was developed incorporating the group operators described in previous sections. The overall EA structure is now summarised;

Step 1. *Input*: initial data input:
- (i) Select the typical initial EA parameter values as shown in Table 2.
- (ii) Input the manufacturing data, with sequence data.

Step 2. *Initial population*: create randomly, an initial population, called *oldpop*.

Repeat

Step 3. *Selection*: Select chromosomes using stochastic sampling without replacement.

(i) Evaluate strings by objective function, fitness function and expected count.
(ii) Create a temporal population, *temppop*, using integer parts of expected count, and fractional parts as success probabilities.

Step 4. *Crossover/recombination*: Apply the group crossover to *temppop* to create a selection pool, *spool*.
(i) Select two candidates for crossover using remainder selection without replacement.
(ii) Apply crossover operator to the two strings; if successful, apply inversion operator, otherwise go to step 5.
(iii) Apply repair mechanism if necessary.

Step 5. *Mutation*: Apply mutation operators to the two offspring and move them to new population.

Step 6. *Replacement strategy*: Replace the less performing strings in old population with selection pool strings.
(i) Compare corresponding chromosomes successively in selection pool and old population.
(ii) Take the one that fares better in each comparison.
(iii) For the rest of the offspring, select with probability 0.54.

Step 6. *Diversification*: Diversify population using the inversion operator if diversity falls below a minimum, H_a.
(i) Calculate diversity H, of the population.
(ii) If $H<H_a$ then diversify until diversity is acceptable.
(iii) Re-evaluate chromosomes in terms of fitness function defined.

Step 7. *New population*: If Advance the new population to the next generation, *oldpop = newpop*
Until (*gen* ≥ *maxgen*).

Part families are identified based on the number of operations required by a part in a cell. Therefore, a part is assigned to a cell where it requires maximum number of machine operations.

Table 2: Typical EA genetic parameter values

EA Parameter	Variable	Value
Number of generations	*maxgen*	100
Population size	*popsize*	10-30
Crossover probability	*pcross*	0.4 – 0.6
Mutation probability	*pmutation*	0.02 – 0.2
Inversion probability	*invprob*	0.04 – 0.2
Chromosome size	*chrom*	Number of machines

5. Results and Discussion

Our proposed EA approach was implemented in Java SE 7. In this section, we provide an illustration of the EA execution, and a comparison of the approach with other algorithms based on known published data sets.

5.1 EA computational results

Numerical results obtained when executing the EA algorithm are presented. Figure 7 illustrates the intermediate stages arrived at as the EA algorithm solves a 25 machine x 40 parts problem from Nair and Narendran (1998). The objective functions are ACFI and the OFI. Assume that the number of cells is four, and $w_1 = w_2 = 0.5$. The ACFI values rose from 23% to 67% after 35 iterations, while the OFI values rose from 16% to 44% after 45 iterations.

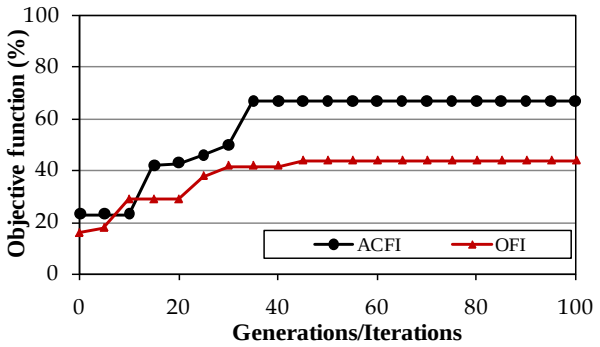

Figure 7: EA objective function for Nair & Narendran (1998) - 25 x 40 problems

Further numerical experiments were done based on an 8 x 20 problem in Nair and Narendran (1998), shown in Table 3. Using typical genetic parameters, simulation results were obtained as shown in Table 4 and 5.

Table 3: An 8 x 20 problem from Nair and Narendran (1998)

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

The results in Table 4 reveal that the EA run provides an improved solution to the problem. Table 5 gives a summary of the improved solution. The cell layout (machine cells) obtained by the EA approach are the same as those obtained from CASE algorithm (Nair and Narendran, 1998) and from CLASS algorithm (Mahdavi and Mahadevan, 2008). Like the CLASS algorithm, EA obtained an improved sequence of machines, with improved machine layout within cells.

Table 4: EA solution for Nair and Narendran (1998) 8 x 20 problem

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

Table 5: The final improved solution of the 8 x 20 problem

Cell	Machines	Parts
C1	1, 3	2, 8, 9, 11, 13, 14, 16, 17, 19
C2	2, 4, 7, 8	3, 4, 6, 7, 18, 20
C3	6, 5	1, 5, 10, 12, 15

To further demonstrate the performance of the EA algorithm, EA results were compared with CASE and CLASS algorithms. Table 6 provides the results of the study. Though machine groups and part families are the same for the three algorithms, ACFI and OFI differ with the CASE solution. The ACFI and OFI values of EA are similar to those obtained from CLASS, showing an improvement of the solution to the cell formation and layout problem.

Table 6: A comparing EA, CASE and CLASS algorithms on a 8 x 20 problem

Cell No.	CASE Solution				CLASS Solution				EA Solution			
	n_c	N_{fc}	N_{tc}	CFI%	n_c	N_{fc}	N_{tc}	CFI%	n_c	N_{fc}	N_{tc}	CFI%
1	9	1	9	11.1	9	5	9	55.6	9	5	9	55.6
2	6	7	18	38.9	6	9	18	50	6	9	18	50
3	5	1	5	20.0	5	2	5	40.0	5	2	5	40.0
N_{flow} = 41												
ACFI (%)				21.0				50.0				50.0
OFI (%)				22.0				39.0				39.0

5.2 Comparison of EA and other algorithms

The Data sets reported by Tam (1988) and Harhalakis et al. (1990) were used for further comparative experiments. Park and Suresh (2003) made a comparative study on known algorithms such as fuzzy ART neural network and conventional clustering methods. Recent related algorithms are CASE designed by Nair and Narendran (1998), and CLASS proposed by Mahdavi and Mahadevan (2008). Table 7 shows the results of the comparative study.

Table 7: A comparison of EA with other approaches

Data set	Size	CLASS			Fuzzy ART			Hierarchical			EA		
		Cells	ACFI	OFI	Cells	ACFI	OFI	Cells	ACFI	OFI	Cells	ACFI	OFI
Tam (1988)	12 X 19	2	65%	50%	2	49%	36%	2	48%	45%	2	65%	50%
Harhalakis et al.(1990)	20 x 20	4	65%	41%	4	42%	34%	4	42%	34%	4	69%	43%
Nair & Narendra (1998)	25 X 40	4	52%	34%	7	38%	27%	8	37%	22%	4	68%	42%
Nair & Narendra (1998)	08 x 20	3	50%	39%							3	50%	39%

According to these results, our EA approach is more preferable than other algorithms found in literature.

6. Conclusions

The integrated cell formation and layout problem is an essential but *NP*-hard problem involving cell formation and machine layout in each cell. The use of sequence data provides valuable additional information on the dominant flow patterns, which form the basis for solving the problem. One of the main challenges, is to extend the application of sequence data and to develop a robust algorithm for solving the joint design and layout problem. In this study, EA was proposed to solve the integrated design and layout problem based on sequence data. The approach has enhanced group chromosome scheme, group crossover operator, group mutation operator, and a chromosome repair mechanism. The operators enable the algorithm to reveal the group structure inherent in a data set, producing comparably high quality solutions. While crossover operator enhances exploration of unvisited points in the potential solution space, mutation exploits the best solution in the near-optimal space. Although increasing the number of cells and/or machines may demand more iteration before convergence to a good solution, the number of parts has no effect on the solution space. EA's parallel mechanism gives the algorithm robustness and effectiveness over a variety of ill-structured input matrices. Thus, the algorithm is quite preferable in large problem situations. Known algorithms in literature were compared with EA approach based on average cell flow and overall flow indices as performance measures. These indices enabled the EA approach to evaluate the cell formation and layout problem in an integrated fashion. Results of the computational study show the utility of the enhanced EA approach. Prospects for further research and application of the proposed EA may be interesting. The grouping concepts in this study can be extended to similar clustering problem domains, such as scheduling and network design.

References

Back, T., 1996. Evolutionary Algorithms in Theory and Practice: Oxford University Press, New York.

Falkenauer, E., 1992. "The grouping genetic algorithms - widening the scope of the GAs," Belgian Journal of Operations Research, Statistics and Computer Science, 33, 79-102.

Filho, E.V.G., and Tibert, A.J. 2006. "A group genetic algorithm for the machine cell formation problem," International Journal of Production Economics.

Gan, X., and Zheng, J. 2011. "Multi-Agent Based Hybrid Evolutionary Algorithm," IEEE Seventh International Conference on Natural Computation 2, 1106-1110

Goldberg, D. E., 1989. Genetic Algorithms: In Search, Optimization & Machine Learning, Addison-Wesley, MA.

Grefenstette, J.J., 1987. "Incorporating problem specific knowledge into genetic algorithms," In L. Davis, Genetic and Simulated Annealing, London: Pitman.

Harhalakis, G. Nagi, R., Proth J.M., 1990. An efficient heuristic in manufacturing cell formation for group technology applications. International Journal of Production Research, 28, 185-198.

Hsu, C.M., and Su, C.T., 1998. "Multi-objective machine-component grouping in cellular manufacturing: a genetic algorithm approach," Production Planning & Control, 9(2), 155-166.

James, T., Brown, E., and Ragsdale, C.T. 2010. "Grouping Genetic Algorithm for the Blockmodel Problem," Evolutionary Computation, IEEE Transactions on, 14(1), 103-111.

Jayaswal, S. and Adil, G. K., 2004. "Efficient algorithm for cell formation with sequence data, machine replications and alternative process routings," International Journal of Production Research, 42, 2419-2433.

Ji-Yang, Q. 2010. "Application of improved simulated annealing in facility layout design," IEEE Proceedings of the 29th Control conference, 5224-5227.

Kaebernick H. and Bazargan-Lari, M., 1996. "An integrated Approach to the Design of Cellular Manufacturing," Annals of the CIRP, 45(1), 421-425.

Kumar, K. R., Kusiak, A. and Vaneli, A., 1986. "Grouping parts and components in FMS," European Journal of Operational Research, 24, 387-397.

Mahdavi, I. and Mahadevan, B., 2008. "CLASS: An algorithm for cellular manufacturing system and layout design using sequence data," Robotics and Computer-Integrated Manufacturing, 24, 488-497.

Mahdavi Mahdavi, I.; Tahami Baher, N.; Teymourian, E.; "A new cell formation problem with the consideration of multifunctional machines and in-route machines dissimilarity - A two phase solution approach," Industrial Engineering and Engineering Management, 2010 IEEE 17th International Conference on, 2010, 475-479.

Nair, G. J. and Narendran, T. T. 1998. "CASE: a clustering algorithm for cell formation with sequence data," International Journal of Production Research, 36, 157-179.

Onwubolu, G.C. and Mutingi, M., 2001. "A genetic algorithm approach to cellular manufacturing systems," Computers & Industrial Engineering, 39, 125-144.

Park, S., Suresh, N.C., 2003. Performance of fuzzy ART neural network and hierarchical clustering for part-machine grouping based on operation sequences. International Journal of Production Research, 41, 3185-3216

Sarker B. R., and Xu, Y., 1998. "Operation sequences-based cell formation methods: a critical survey," Production Planning & Control, 9, 771-783.

Tam, K.Y., 1988. "An operation sequence based similarity coefficient for part family formations," Journal of Manufacturing Systems, 9, 55-68.

Venugopal, V. and Narendran, T.T., 1992. "A genetic algorithm approach to the machine-component grouping problem with multiple objectives," Computers and Industrial Engineering, 22 (4), 469-480.

Won, Y. and Lee, K. C., 2001. "Group technology cell formation considering operation sequences and production volumes," International Journal of Production Research, 39, 2755-2768.

Wu, T.H., Yeh, J.Y, and Chan, C.C., 2009. "A hybrid Tabu Search Algorithm to Cell Formation Problem and its Variants," World Academy of Science, Engineering and Technology 53, 1090-1094.

Yin, Y., and Yasuda, K.. 2004. "Reconsidering generalized similarity coefficient via a sequence ratio," International Journal of Industrial Engineering – Theory, Applications and Practice, 11, 140-150.

Yin, Y., Yasuda K., and Hu, L. 2005. "Formation of manufacturing cells based on material flows," International Journal of Advanced Technology, 27,159-165.