A novel hybrid DCMA-SSA paradigm for the multi-objective Cell Formation Problem

Article *in* Interciencia · August 2020

3 authors:

Nagaraj G
Sethu Institute of Technology

Dr. Vinayagar
Sethu Institute of Technology

Kps Paramasamy
Sethu Institute of Technology

A novel hybrid DCMA-SSA paradigm for the multi-objective Cell Formation Problem

G.Nagaraj*.
Department of Mechanical Engineering.
Sethu Institute of Technology.
Virudhunagar, India
Email: nagarajsasi@gmail.com
*Corresponding Author

Dr.A.Manimaran.
Department of Mechanical Engineering.
M.N.S.K. College of Engineering.
Pudukottai, India
Email: mullaimaran@rediffmail.com

Dr.K.Vinayagar,
Department of Mechanical Engineering.
Sethu Institute of Technology.
Virudhunagar, India
Email: kvinayagar@sethu.ac.in

S.Paramasamy.
Department of Mechanical Engineering.
Sethu Institute of Technology.
Virudhunagar, India
Email: sparamasamy@sethu.ac.in

ABSTRACT: Cell Formation Problem (CFP) is a great combinatorial optimization problem. It comprises grouping of machines and parts into manufacturing cells for processing the parts. For this, we propose multi-objectives of CFP, with the objective to minimize (1) intercellular movements and (2) Cell Load Variation (CLV) in a given period of time. These multi-goals in CFP is assessed by utilizing mathematical modeling and soft computing techniques. Here a novel hybrid optimization algorithm combination of Discrete Cauchy Mutation Algorithm (DCMA) with Salp Swarm Algorithm (SSA) is presented. DCMA is the updated model of genetic algorithm and the hybridization of DCMA-SSA is maintained leading and gradually more powerful for engineering design problems compared with the existing algorithms. The performance of the hybrid algorithm is compared with the standard benchmark problems. The computational time of the proposed work, performs higher effectiveness while implementation.

Keywords: Cell Formation Problem, intercellular movements, cell load variation, DCMA-SSA

1. Introduction

Cellular Manufacturing is a production system (CMS) where comparative parts are grouped into part families and different machines are fixed into machine cells so as to develop the cost-capability of large scale manufacturing and adaptability of job shop manufacturing (Izquierdo et al., 2016; Mukattash et al., 2018 and Hazarika and Laha, 2015). In computerized batch type production frameworks, machine-part CFP has long drawn consideration of analysts. Machines are doled out to the cells to process at least one section families with the aim that every cell is worked separately and between the cellular movements are limited. The CFP is initiated by Burbidge (1963) and extended through the (Zhao & Wu, 2000; Ghosh et al., 2011; Javaid et al., 2014; Noktehdan et al., 2010) by presentation of principal strategy for Production Flow Analysis. For a considerable length of time, numerous strategies for the CFP have been reported.

The majority of the machine-part cell formation problems are NP-hard problems; along these lines, optimization algorithms yield all-inclusive perfect solution in a restrictive calculation time (Delgoshaei et al., 2016; Rajesh et al., 2006; Ghosh and Dan, 2012; Tunnukij. and Hicks, 2009). The computational intractability of the problem utilizes suitable heuristic methodologies (Dimopoulos 2006; Adinarayanan et al., 2018). Mathematical programming based methodologies have the capability of explaining the CMS design problem ideally and this is the reason that such methodologies have been broadly used to take care of the CFP (Baykasoglu, 2001; Selim, 2002; Xambre and Vilarinho, 2003; Elbenan and Ferland, 2012).

In traditional, techniques utilized in CM are classification and coding. Few methodologies have been proposed to deal with CFP and these techniques to develop the cell formation are clustering, genetic algorithm, neural network based methodologies, numerical programming based methodologies and heuristic based methodologies. In this manner the above said algorithms can't ensure global optimality of the attained solutions as for a variable number of production cells. As a rule the computational time for this model are very long or memory constraints are surpassed and the optimal solutions can't be found. The fundamental objective of CFP is reducing the cell and intra-cell transportation costs, grouping efficiency, grouping efficacy, and so on. The proposed investigation presents a novel methodology in CFP which has the goal of least intercellular movements and cell load variation. To approve these goals a new methodology is proposed and simulate with a hybrid strategy.

The rest of the paper is structured as follows: Section 1 presents the introduction part which introduces about cell formation problem, section 2 reviews recent literatures related to proposed topic. Section 3 presented the methodology section and the section 4 depicts the results and discussion finally the conclusion part concludes the study.

2. Related Research

Nagaraj et al., (2020) suggested a hybridized monarch butterfly optimization and firefly algorithm for the solution of multi-objective cell formation problem to large sized problems with the short measuring of time. Experimental results performs superior to the existing known algorithms. Hosseinabad and Zaman (2020) presented a research structure to the researchers, academicians and industrial experts regarding the lean progression in cellular manufacturing through the complete reviews. They also presented the execution within an organization, and recommended prospective research part enhancement to the cellular manufacturing. Rabbani et al., (2019) provided a modern multi-objective conceptual model for the complex cellular manufacturing system (DCMS) by considering the machine efficiency and alternate method routes. In this dynamic model, the solution provided for the integrated cell formation (part / machine) and the assignment of cells to the operators. The first target lowers the costs of the DCMS to a minimal. The second goal optimizes the use of work and the third objective function ultimately obtains a minimum value of the working load variance between different cells. Due to the NP-hard nature of the cellular manufacturing problem, the problem is initially validated by the GAMS software in small-sized problems, and then the model is solved by two well-known meta-heuristic methods including non-dominated sorting genetic algorithm and multi-objective particle swarm optimization in large-scaled problems.

Danilovic & Ilic (2019) recommended a new approach to use the specificities of the input inst new approach is to use the specificities of the input instances to narrow down the feasible set, and thus increase the efficiency of the optimization process. The time efficiency of the proposed algorithm is at least an order of magnitude better than the efficiency of the most efficient reported algorithms. Paramasamy et al., (2019) developed a genetic algorithm for concurrent formation of part families and machine cells for CMS with the objective of minimization of exceptional elements in a sheet metal industry.

Ulutas, (2018) has examined the Clonal Selection Algorithm (CSA) with a novel encoding structure that was proficient to tackle real-sized problems. Proposed CSA was tested on 67 problems. CSA gets the equivalent 63 best-known ideal solutions, gives a solution for the 3 of the notable test problem and another solution for the biggest test problem i.e. with 50 machines 150 sections that were impractical to be illuminated by the mixed-integer linear programming model because of the high computational multifaceted nature.

Hazarika and Laha, (2017) introduced a genetic algorithm for solving benchmark problems. The authors decided the optimal processing course and adjusted machine cells (to limit cell load variation) fuse with parts volume, process succession for least intercellular movements of parts. Mahdavi et al., (2009) has proposed nonlinear terms and integer variables can't be tackled for real sized problems productively because of its NP-hardness. To solve the model, a genetic algorithm approach was proposed. Numerical examples demonstrated that the proposed strategy was proficient and compelling in searching for ideal solutions. The outcomes also demonstrated that the proposed methodology performs well regarding group efficacy compared with the notable existing cell formation strategies.

Noktehdan et al. in 2016 had proposed a practically new algorithm named League Championship Algorithm for grouping adaptation they utilized it to comprehend benchmarked cases of CFP and acting as a gathering problem. In addition, a true mechanical case was given to indicate how the proposed algorithm functions. The authors demonstrated their result that, grouping league championship algorithm (GLCA) algorithm attains high solution quality and also compared the results with some other algorithms. In recent times, some authors have contended that it is conceivable to cause genetic algorithm for taking care of optimization issues to perform better by considering a nearby improvement process over each new individual produced in a population. Zohrevand et al., (2016) displayed bi-objective stochastic model tries to limit the complete expense of machine acquirement, machine migration, between cell moves, additional time use, specialist employing/laying-off, and worker moves between cells; while the second target capacity maximizes work use of the cellular manufacturing system. A hybrid Tabu Search-Genetic Algorithm was proposed whose quality was approved to acquire optimally and close optimal arrangements through led experimental results.

Nagaraj et al., (2015) provided a case study for identifying bottlenecks in the assembly shop of a cellular manufacturing automobile company and removing bottlenecks through the reduction of cycle time for the enhancement of productivity. Nagaraj et al., (2015) examined three array-based clustering algorithms, named rank order clustering (ROC), rank order clustering-2 (ROC2) and direct clustering analysis (DCA) for the CFP, with a real-time instance to express the effectiveness of various clustering algorithms. The most effective method is selected and used to build the cellular manufacturing system. Manimaran et al., (2014) recommended a modified ART1 (MART1) for solving cell formation problem using production factors such as operation sequences and operation sequence with production volume. Manimaran et al., (2013) applied the Back Propagation Network (BPN) based algorithm to form the machine cells and component grouping for minimizing the exceptional elements and bottleneck machines.

By reviewing different literatures; various techniques and algorithms are presented in last decades related to cell formation problem (Yang et al., 2016; Liu et al., 2008; Dimopoulos, 2006; Zhou et al., 2019; Neufeld, 2019).

To obtain an adequate solution, we introduced hybrid algorithms like DCMA-SSA for solving cell formation problem. The hybrid model is shown to perform better than other benchmarking approaches both in solution effectiveness and efficiency, even in large-sized problems.

3. Cell Formation Problem

One of the basic problems in cellular manufacturing framework is the cell formation in which formation of part families that processing on a set of machines called machine cells. Every cell is equipped for fulfilling every one of the prerequisites of the part family appointed to it. Then again, the objective of CFP in our work is reducing cell load variation and intercellular flows. To solve this target in the cell formation problem, we present a novel hybrid approach of DCMA-SSA technique which is implemented in MATLAB software.

Let us taken a 0-1 part machine incidence matrix, the cell formation problem can be planned as a block diagonalization issue. We are looking for an appropriate arrangement of rows and columns of the machine part incidence matrix to make a block diagonal matrix showing part families and machine cells.

Table 1 shows a machine-part incidence matrix important to an issue with five parts and five machines. In this matrix, no blocks can be watched straightforwardly. Nonetheless, after diagonalization process, the block diagonal matrix is acquired as outlined in Fig.1b. Consider a production flow comprising seven parts, five machines and 14 activities (incidence matrix positions with an estimation of 1).

Table 1(a) and (b) depicts the input matrix, and possible solution matrix with the formation of two cells (families) recognized by the shaded area. In this example, we get two cells, where the first cell contains machine {2, 3, 5} and parts {1, 3, 7}, and the second cell has machines {1, 4} and parts {2, 4, 5, 6}.

Table 1 (a) Initial matrix

	m1	*m2*	*m3*	*m4*	*m5*
p1	0	1	1	0	1
p2	1	0	0	1	0
p3	0	1	1	0	0
p4	1	0	0	1	0
p5	1	0	0	0	1
p6	1	0	1	1	0
p7	0	0	1	0	1

Table 1 (b) After rearrangement

	m2	*m3*	*m5*	*m1*	*m4*
p1	1	1	1	0	0
p3	1	1	0	0	0
p7	0	1	1	0	0
p2	0	0	0	1	1
p4	0	0	0	1	1
p5	0	0	1	1	0
p6	0	1	0	1	1

3.1 Mathematical modeling

The objective of this DCMA-SSA technique is the minimization of part intercellular movements of and cell load variation. Besides, maximizing the performance can improve the proficiency of the cell formation problem. To limit the variation of the total load inside the cell, this helps softening the variation among the parts inside every cell, prompting a work in procedure minimization inside every cell. These goals can mathematically formulate as pursues:

Indices

i Index for parts; $i = 1,2,....p$

j Index for machines; $j = 1,2....m$

k Index for cells; $k = 1,2,.....c$

m Total number of machines

p Total number of parts

c Total number of cells

Notations

$$g_{ik} \begin{cases} 1 & if \;\; part \;\; i \;\; is \;\; in \;\; cell \;\; k \\ 0 & otherwise \end{cases}$$

$$h_{jk} \begin{cases} 1 & if \;\; machine \;\; j \;\; is \;\; in \;\; cell \;\; k \\ 0 & otherwise \end{cases}$$

$$q_{jk} \begin{cases} 1 & if \;\; machine \;\; j \;\; is \;\; assigned \;\; to \; cell \;\; k \\ 0 & otherwise \end{cases}$$

y_{ij} Manufacturing relation between machine j and part i $\begin{cases} y_{ij} = 1; & if \ \ t_{ij} > 0 \\ 0 & otherwise \end{cases}$

w_{ij} The workload on machine j induced by part i $w_{ij} = \left(\frac{t_{ij} \times R_j}{T_j} \right)$

b_{ij} The average intracell processing times for part i in cell k; $b_{ij} = \left(\frac{\sum_{i=1}^{m} g_{ik} h_{jk} w_{ij}}{\sum_{i=1}^{m} g_{ik}} \right)$

δ The weight factor to balance the intercell flows and cell load variation; $0 \le \delta \le 1$

Cell Load Variation

$$F_1 \Rightarrow L_c = \sum_{i=1}^{m} \sum_{k=1}^{c} \sum_{j=1}^{p} g_{ik} h_{jk} \left(w_{ij} - b_{jk} \right)^2 \quad (1)$$

Intercellular flows

$$F_2 \Rightarrow I_f = \sum_{k=1}^{c} \sum_{i=1}^{m} \sum_{j=1}^{p} R_j y_{ij} \left(1 - g_{ik} h_{jk} \right) \quad (2)$$

The sum of objective function can be denoted as

$$F \Rightarrow min(F_1 , F_2) \quad (3)$$

Constraints:

$$\sum_{k=1}^{c} g_{ik} = 1; \qquad for \ i = 1,2,..........p \quad (4)$$

$$\sum_{k=1}^{c} h_{jk} = 1; \qquad for \ \ j = 1,2,..........m \quad (5)$$

$$\sum_{i=1}^{m} g_{ik} \ge 1; \qquad for \ k = 1,2,..........c \quad (6)$$

$$\sum_{k=1}^{c} h_{jk} \ge 1; \qquad for \ k = 1,2,..........c \quad (7)$$

$$g_{ik}, h_{jk} \in \{0,1\} \ \ for \ i = 1,2,...... p; \ j = 1,2,...... m; k = 1,2,...... c; \quad (8)$$

Table 2 Benchmark problems from existing literature

S. No.	*Problem Source*	*No. of machines*	*No. of parts*	*No. of cells*
1	Kao & Lin (2012)	7	10	3
2	Chang et al. (2013)	10	10	3
3	Arkat et al. (2007)	17	30	4
4	Hanxin feng et al.	5	5	2
5	Pandiyan et al. (2000)	25	40	4
6	Kusiak & cheng (2013)	7	11	3
7	Saedi et al. (2000)	15	25	3
8	Ghosh, Dolai, & Dan	4	5	2
9	Murugan et al. (1999)	7	8	2
10	McAuley (1998)	7	10	3
11	Chan and Milner (1982)	10	15	3
12	Chu & Hayya (1991)	9	9	2
13	Stanfel (1985)	14	24	5
14	McCormick and Schweitzer (1972)	16	24	6
15	Srinivasan et al (1990)	16	30	4
16	King (1980)	16	43	5
17	Karpathi & Suresh	20	35	6
18	Mosier and Taube	20	20	5
19	Kumar et al. (1986)	20	23	5
20	Harhalkis et al. (1969)	20	20	4

As indicated by the above model, for a given machine-part incidence matrix w_{ij} and a given weight factor, the result we need to get is the quantity of cells c, machine–cell membership g_{ik} and part-cell membership h_{jk} after cell formation.

- Equation (1) demonstrates the computation of the cell load variation and conditions (2) the intercellular movements.
- Equation (3) is our target function for CFP, which is adjusted by a weight factor.
- Equation (4) and (5) guarantee that each machine and part must be appointed into one cell.
- Equation (6) and (7) guarantee that every cell must contain in any event one machine and one part, separately.
- Equation (8) expresses that g_{ik} and h_{jk} is 0-1 binary decision factors.

Mathematical models of the CFP procedure may derive computational difficulties and may not be resolvable using commercial optimization software for medium-to-broad estimated problems. As such, proficient heuristic methodologies are required to agreement with the problems of greater sizes. Here we develop a hybrid heuristic strategy dependent on DCMA-SSA to comprehend the proposed model. Diverse benchmark problems with different sizes are shown in table 2.

3.2 Discrete Cauchy Mutation Algorithm (DCMA)

DCMA plays out similar general procedures of Genetic Algorithm endeavoring to mimic biological evolution, while DCMA would endeavor to Cauchy's mutation and furthermore taken the discrete formation of the population in genetic algorithm. As a rule, GA has been created by Holland and Goldberg is a stochastic based global search optimization algorithm guided by the regular development and genetics standards. It introduces with a lot of solutions, known as initial population and after that executes sequentially selection, reproduction, and crossover and cauchy mutation activities for a fixed number of iterations as a stopping criterion. Fitness condition: The objective function could be evaluated on account of the fitness function. The fitness function $F(c)$ is evaluated as minimizing the intercellular movements and cell load variation. It is deliberated in below equation (1).

$$F(c) = \left\{ \min \begin{pmatrix} \textit{Intercellular movements} \\ \text{Cell load variation} \end{pmatrix} \right\} \qquad (9)$$

Machinery of DCMA: The DCMA makes unique genetic operators, which are appropriate for chromosomes with three segments of parts, machines, and cells. It is fit for managing a variable number of cells. Every iteration (additionally alluded to as generation or group of solutions) is comprised of chromosomes. Every chromosome is thus comprised of individual genes. These genes are encodings of the design variables that are utilized to assess the function being upgraded. In every cycle of the search procedure, the framework has a fixed populace of chromosomes that speak to the present solutions for the problem.

Generate Population: In the initial step, cauchy mutation instates the population with the generation of a set of beginning random solution. Deciding the best possible population size is a noteworthy choice in cauchy mutation implementation. If the chosen number is extremely small, we are not able to get a decent arrangement. Then again, if it is excessively enormous, it might consume a lot of CPU time to acquire a superior solution. An exceptional method was created in this exploration to produce a random initial population while constraints 2 and 3 are fulfilled, for example, each machine and each part ought to be doled out to just a single cell. A chromosome comprises of a series of genes and there is one gene for each machine, the value of the quality equivalents the quantity of the cell to which the comparing machine is allotted. Therefore, the quantity of qualities in the chromosome will constantly rise to the absolute number of machines. Note that the part and machine bits of the chromosome are fixed long dependent on the size of the problem.

For example: Let us assume there are 5 machines numbered from (M1, M2, M3, M4, M5), 7 Parts (P1, P2, P3, P4, P5, P6, P7) and two cells recognized by whole numbers C1, C2 then the series of genes is (2, 1, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1) gives out a partition of 5 machines that machine 1 is in cell 2, machine 2 is in cell 1, and so forth.., these chromosome portrayal is displayed in beneath table 3:

Evaluating Fitness: In this proposed model, the fitness is assessed based on condition (3). The fitness speaks by limiting the intercellular movements and cell load variation. Calculation of fitness estimation of every chromosome in the population is a model of the determination procedure to evaluate the high probability of choosing the candidate solution for the next iteration. The larger fitness value is having a higher probability of survival for the next generation.

Table 3 Chromosome representation

Machine /Parts	*M1*	*M2*	*M3*	*M4*	*M5*	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*	*P7*
Gene	0	1	2	3	4	5	6	7	8	9	10	11
Cells	2	1	1	2	1	1	2	1	2	2	2	1

Operation Selection: The selection of individuals to create progressive generations assumes a critical job in a genetic algorithm. A discrete selection is performed dependent on the individual's fitness to such an extent that the better individuals have an expanded shot of being chosen. Holland (1975) built up the roulette wheel, which was the primary determination technique. Genetic operators work to cause the populace to develop.

Generate a new set of the population: Very few quantities of chromosomes are chosen with more noteworthy fitness values (parent) from the populations through which produce another set of solutions (offspring).

Crossover operator: Crossover is the genetic operator wherein two parent chromosomes are exposed to deliver two new offspring. In the DCMA process, two-point crossovers are utilized; the initial two chromosomes from the population are chosen based on their higher fitness value and permitted to partake in the crossover operation. From the referenced model, we expect two parents (parent 1 and parent 2) are taken at a randomly chosen location and they produce another two new chromosomes state, offspring1 and offspring 2.

Cauchy mutation Process: Mutation operator in the GA is required to present assorted variety in the population starting with one generation then onto the next so as to dispose of stalling out at local optimum. If there should arise an occurrence of better fitness values these offspring would supplant the weakest solutions in the population. The code is created so that it would exclude any illegal child into the population.

On account of Cauchy mutation, the random variable Z is Cauchy distribution by Choi. and Ahn (2018), Wang et al (2007), Wu & Law (2011). Iqbal et al (2010). The explanation behind utilizing Cauchy mutation operator is to expand the probability of getting away from a local optimum. The one-dimensional Cauchy density function-focused at the starting point is characterized by:

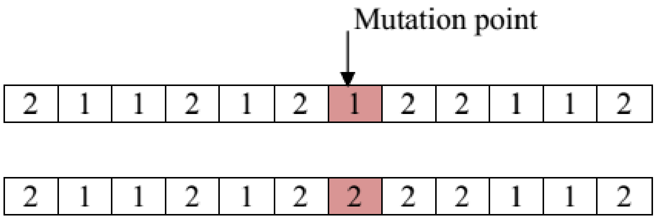
$$f(x)=\frac{1}{\pi}\left[\frac{h}{h^2+z^2}\right], \quad -\infty<z<\infty \qquad (10)$$

Where $h>0$ is a scale parameter; the Cauchy distribution function of DCMA is calculated by equation (11).

$$F_h(x)=\frac{1}{2}+\frac{1}{\pi}arctan\left(\frac{z}{h}\right) \qquad (11)$$

The cauchy mutation is used to mutate the individuals according to equation (9 and 10). Mutation is performed on the basis of pre-determined mutating probability.

Figure 1 Mutation Process (before and after)



To get the feasible optimum solution, repeat the above procedure for a specific number of iterations. The number of iterations is normally relies on the size of the problem. As the arrangement space expands, the DCMA will require a higher number of generations to conceivably achieve a convergence point. Population size may differ contingent upon the application. The number of iterations must be set to enable the DCMA to finish the convergence procedure.

3.3 Salp Algorithm Swarm (SSA)

SSA is inspired by the swarming behavior of salps when investigating and rummaging in oceans. The salp chain attempts to find the best area of food through the searching system with the help of a pioneer salp, with the others as followers. The circumstance of all salps is secured in a two-dimensional framework called D. It is expected that there is a food source F_s acquired the search space as the swarm's goal. To update the circumstance of the leader the accompanying condition is proposed:

$$G_j^1 = \begin{cases} F_j + R_1\left[\left(U_j - L_j\right)R_2 + L_j\right] & R_3 \geq 0 \\ F_j - R_1\left[\left(U_j - L_j\right)R_2 + L_j\right] & R_3 < 0 \end{cases} \quad (12)$$

Where, F_j shows the position of the first parts (leader) in j^{th} cell, the position of the machines in j^{th} cell is symbolized as F_i; the upper bound and lower bound is indicated as U_j *and* L_j; $R_1, R_2, and\ R_3$ indicates random number randomly generated in the interval of [0,1]. The salp leader updates its siting according to the food source. The coefficient C_1 is the most basic parameter in SSA since it changes of exploration and exploitation portrayed as pursues:

$$C_1 = 2e^{(-4l/L)} \quad (13)$$

Where l is the present iteration and L is the greatest number of iterations. To update the situation of the parts, Newton's law of motion is used. The disparity between iterations is equivalent to 1, and considering the underlying speed as 0, this condition can be communicated as pursues:

$$S_j^i = \frac{1}{2}\left[S_j^i + S_j^{i-1}\right] \quad (14)$$

Where $i \geq 2$ and S_j^1 demonstrates the situation of i^{th} follower salp in j^{th} dimension. With condition (14), the salp chains can be simulated. It ought to be noticed that the food source will be refreshed among optimization since the salp chain is incredibly liable to find a superior solution by investigating and abusing the space around it. Along these lines, the salp chain can possibly move towards the worldwide ideal that changes throughout cycles.

3.4 Hybrid (DCMA-SSA)

The hybridization approach is by finding the optimal solutions i.e. minimizing the cell load variation and intercellular movements by which constructs the concept combining certain steps in DCMA and SSA to perform DCMA-SSA. This hybrid approach performs that comparing both algorithm solutions and finally selecting the best solution among two.

The best solution achieves optimal fitness with high efficiency and low computational time. The optimal placement of machines in cells is resolved on the optimal position of the salps or mutating the point by virtue of expanding iterations.

4. Result Analysis

In this section, result for benchmark problems are examined with the DCMA-SSA and validated. Here, we select diverse problem size like small size problems and large size problems. The proposed technique is actualized in MATLAB programming with the most recent version of 2016a and completed in the processor of Intel center i5 with 8 RAM.

Table 4 Intercellular movements for small-sized benchmark problems

S.No.	*Problem size*	*Intercellular movements*				*Computational time*
		EA	*SSA*	*DCMA*	*DCMA-SSA (proposed)*	
1	7x10	10	11	9	7	0.75
2	5x5	15	13	10	5	0.15
3	8x20	13	15	11	8	8.58
4	7x8	19	22	18	15	0.15
5	4x5	15	13	11	9	1.75
6	7x11	20	25	21	18	5.88
7	9x9	19	21	18	15	3.03
8	8x20	23	28	26	20	0.20
9	10x15	29	27	25	22	11.73
10	16x24	20	19	17	14	0.14

The parameter utilized for approving DCMA-SSA is given as pursues in table 5. The parameters required to run the algorithm are population size, number of generations, number of iterations and mutation probabilities. These parameters have an essential job in the presentation of the DCMA-SSA. The proposed hybrid algorithm accomplishes ideal solutions which have minimum cell load variation and minimum intercellular movements. Table 4 shows the Intercellular movements for small-sized benchmark problems with the proposed algorithm.

Table 5 Parameters used in DCMA

Parameter	*Value*
Population size	200
Chromosome size	Depends on Parts and machines
Crossover operator	Two-point crossover
Selection type of crossover	Roulette Wheel selection
No of generations	Varies
Mutation Probability	0.1-1

Table 6 Intercellular movements for small-sized benchmark problems

S.No.	*Problem size*	*Intercellular movements*				*Computational time*
		EA	*SSA*	*DCMA*	*DCMA-SSA (proposed)*	
1	5x7	10	11	9	7	0.75
2	5x7	15	13	10	5	0.15
3	5x18	13	15	11	8	8.58
4	6x8	19	22	18	15	0.15
5	7x11	15	13	11	9	1.75
6	7x11	20	25	21	18	5.88
7	8x12	19	21	18	15	3.03
8	8x20	23	28	26	20	0.20
9	8x20	29	27	25	22	11.73
10	10x10	20	19	17	14	0.14

Table 7 Cell load variation for small-sized benchmark problems

S. No.	*Problem size*	*Cell load variation*				*Computational time*
		EA	*SSA*	*DCMA*	*DCMA-SSA (proposed)*	
1	7x10	6.1	5.9	5.67	5.63	0.25
2	5x5	2.2	2.17	2.13	2.13	0.36
3	8x20	3.56	3.4	3.37	3.35	0.14
4	7x8	8.54	8.4	8.25	8.25	0.85
5	4x5	9.35	9.3	9.12	9.09	0.74
6	7x11	2.49	2.6	2.49	2.48	0.62
7	9x9	3.14	3.24	3.15	3.14	0.51
8	8x20	7.3	7.4	7.29	7.27	0.41
9	10x15	11.39	11.5	11.4	11.37	0.39
10	16x24	4.6	4.7	4.5	4.41	0.55

Table 6, 7 shows the intercellular movements and cell load variation for ten benchmark problems. The proposed result is compared with individual EA, SSA, and DCMA. The computational time for each benchmark problems is analyzed, the intercellular movements and cell load variation of the solution reached by the proposed method is optimal than that of existing algorithms (EA, SSA, and DCMA) (Gonçalves, J.F. and Resende, M.G., 2004). Here, the minimum value of intercellular movements reached in problems 2, 1, 3, and 5; in cell load variation, the minimum value achieves in

 ISSN: 0378-1844

the CFP as problems 2, 6, 7, 3 and 10. Moreover, the proposed hybrid algorithm (DCMA-SSA) has reduced computational time. The flowchart for the proposed hybrid approach DCMA-SSA as shown in figure 2.

Figure 2 Flowchart for the proposed hybrid approach

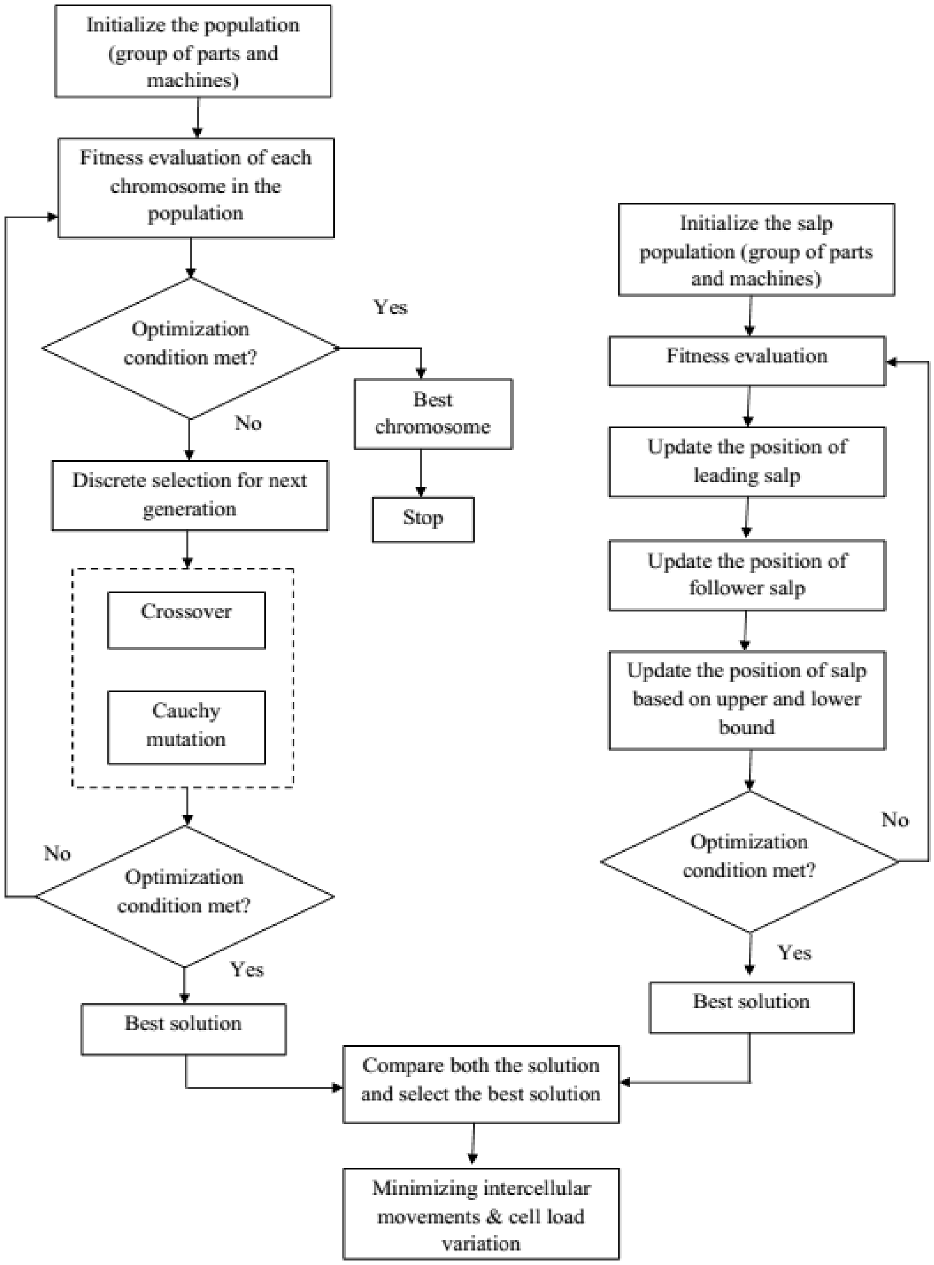


Table 8 Intercellular movements for large-sized benchmark problems

S. No.	*Problem size*	*Intercellular movements*				*Computational time*
		EA	*SSA*	*DCMA*	*DCMA-SSA (proposed)*	
1	17x30	25	21	19	16	0.49
2	14x24	15	15	10	10	0.44
3	25x40	29	25	23	21	1.50
4	16x30	37	33	33	31	0.59
5	16x43	54	49	51	45	0.89
6	18x24	26	17	15	10	5.19
7	20x20	29	32	29	24	0.55
8	20x23	51	49	49	45	0.60
9	20x35	63	59	55	51	0.73

Table 9 Cell load variation for large-sized benchmark problems

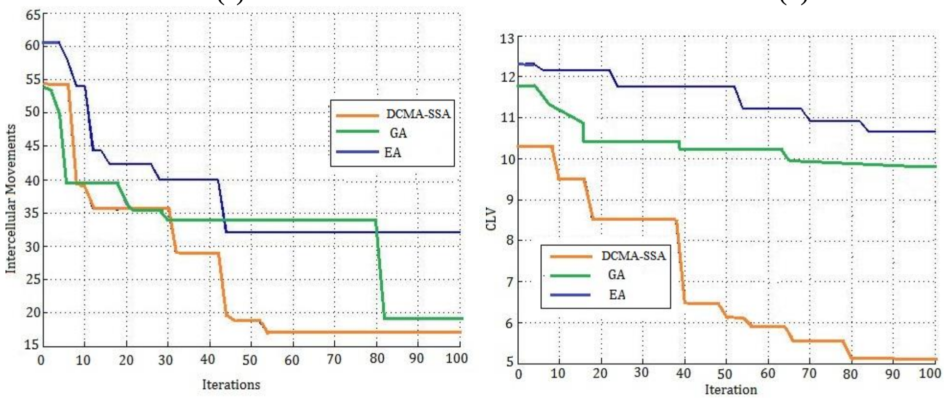
S. No.	*Problem size*	*Cell load variation*				*Computational time*
		EA	*SSA*	*DCMA*	*DCMA-SSA (proposed)*	
1	17x30	20.45	21.55	21.54	20.45	0.99
2	14x24	19.65	17.36	16.25	16.25	1.15
3	25x40	15.66	11.52	11.41	10.78	2.56
4	16x30	40.15	38.44	38.41	38.41	5.26
5	16x43	20.45	20.63	19.74	19.74	4.65
6	18x24	17.45	16.32	15.78	15.62	9.62
7	20x20	28.63	26.33	25	24.99	9.54
8	20x23	18.45	17.18	18.12	17.18	8.41
9	20x35	33.56	31.84	32.65	31.84	9.88

Table 8 and 9 shows the performance measures of the validation result based on large sized benchmark problems. In these types of problems, problem number 2 gives an optimal solution with less amount of computational time. In the validation part, the population of the DCMA-SSA expands if the CFP size is increased. The use of DCMA-SSA will discover the local and global optimum effectively and simultaneously.

Figure 3 shows the performance of fitness function based on iterations for benchmark problem 8 x 20. The convergence graph compares the proposed method into existing algorithms EA and GA. Here we take 100 iterations for the analysis; for the comparison of all the three optimization techniques proposed DCMA gives an optimal solution i.e. minimum fitness values. The result depicts that the effectiveness of the algorithm decides the number of machines and parts into a number of cells.

Figure 3 Objectives based on iterations for benchmark problem (8 x 20) (a) Intercellular movements (b) Cell Load Variation

(a) (b)



Similarly, CLV also varies based on the iteration for problem size (8 x 20). DCMA produces minimum CLV compared to the existing one. It achieves the optimal fitness in the 80th iteration. To minimize the intercellular movements can be achieved by optimal material transfer within the cells based on the function of the machines. Cell load variation depicts the system imbalance, to minimize these over-allocating problems we have to choose the optimal machine loads that can decrease the pressure of loads.

Figure 4 Fitness function based on generations

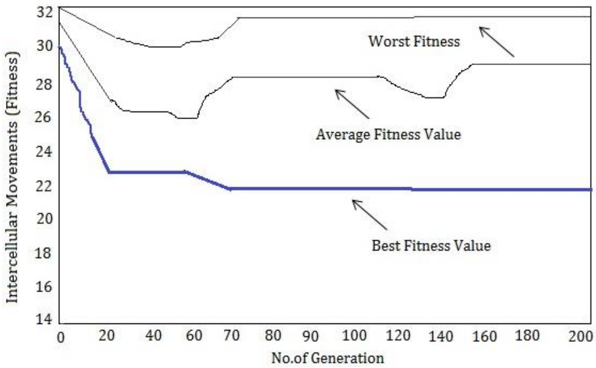


Figure 4 shows the fitness function based on a number of generations. If machines are grouped into the same cells, copy one of the parents as one child; another child is created randomly with the initialization operation. And then break out from the crossover operator; according to our test, we find that the repetitions of the same individual with better fitness will be overfull as the generations grow, and those excess repetitions will induce trapping into local optima.

The best fitness means the optimal solution achieved and the average fitness shows the solution with nearby optima moreover the worst fitness means the solution with maximum fitness. The graph performs the best fitness value as the minimum and shows the average and worst fitness values. Our method is capable of dealing with CFP more efficiently and flexibly without predetermining the number of cells.

Figure 5 Computational time analysis for (a) small-sized and (b) large-sized problems

(a) (b)

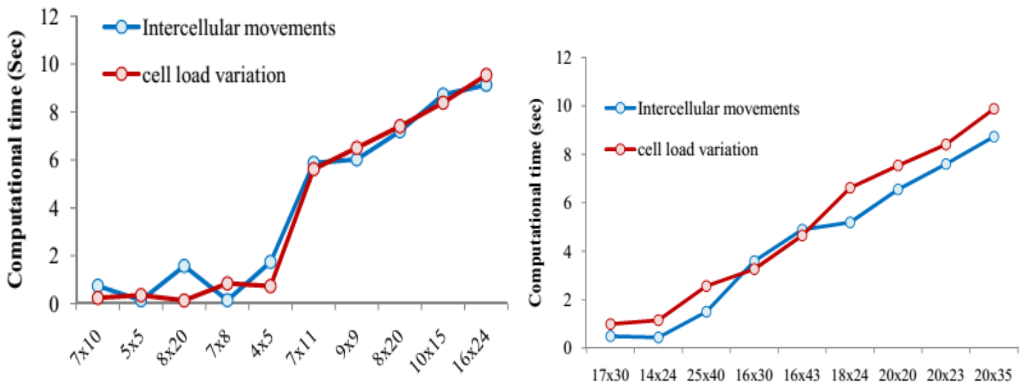


Figure 6 Computational time analysis by comparing existing techniques

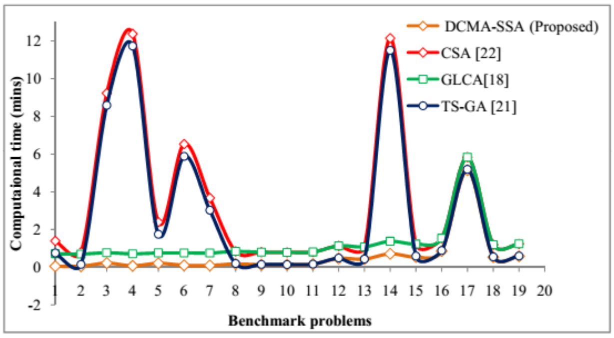


Figure 5 and 6 shows the computational time analysis for small and large-sized benchmark problems. The computational time may vary based on the benchmark problem size. Here, we take different sized benchmark problems to find the objectives. The figure shows the computational time for each benchmark problems with respect to the proposed hybrid approach. Large problems get more running time compared to small-sized problems. Figure 6 shows the computational time for all the benchmark problems and also compares the proposed DCMA-SSA model into existing techniques' like CSA, GLCA, TS-GA. Compared to existing algorithm proposed algorithm finds the objective in minimum computational time with optimal solution.

5. Conclusion

In the novel methodology, CFP is solved with a creative hybrid algorithm, for example, DCMA-SSA dependent on the goal of minimum intercellular movements and cell load variation. Some benchmark problems are utilized to comprehend these targets; the benchmark problems containing large-sized and small-sized problems. DCMA deals with record of GA which resolves with genetic operators, for example, Cauchy mutation operator. The cauchy mutation operator is essentially used to keep up decent variety in the population and a low probability is set to this parameter. The presentation of the proposed hybrid algorithm is efficiently assessed in correlation with other existing techniques on standard benchmark testing problems. The performances of the proposed algorithms are superior to or focused on the notable existing algorithms (EA, and GA). The proposed hybrid technique gives an optimal solution which empowers ideal fitness for every benchmark problems (minimum intercellular movements and minimum cell load variation). In the future, the researchers will improve the CFP data, in actuality, circumstances. Since the essential intelligent software right now created to take care of test problems, can possibly be utilized for mechanical applications. Some benchmark problems can't be solved with minimum computational time because of the NP-fulfillment of the model; therefore, the utilization of meta-heuristics for tackling such a difficult problem, and to acquire solutions that are more proficient and furthermore recommended for future research.

Reference

Adinarayanan, A., Uthayakumar, M. and Prabhakaran, G., (2018) ‘Machine cell formation using simulated annealing algorithm in cellular manufacturing system’, *IJCAET,* Vol. 10, No. 2, pp.111-125.

Baykasoglu, A., Gindy, N. N. Z., & Cobb, R. C. (2001) ‘Capability based formulation and solution of multiple objective cell formation problems using simulated annealing’, *Integrated Manufacturing Systems*, Vol. 12, No. 4, pp.258–274.

Choi, T.J. and Ahn, C.W., (2018) ‘Accelerating differential evolution using multiple exponential cauchy mutation’, *In Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp.207-208.

Danilovic M, Ilic O, (2019) ‘A novel hybrid algorithm for manufacturing cell formation problem”, *Expert System Application*, Vol.135, pp. 327–350.

Delgoshaei, A., Ariffin, M., Baharudin, B. and Leman, Z., (2016) ‘A new method for decreasing cell-load variation in dynamic cellular manufacturing systems’, *International Journal of Industrial Engineering Computations*, Vol. 7, No. 1, pp.83-110.

Dimopoulos, C. (2006) ‘Multi-objective optimization of manufacturing cell design’, *International Journal of Production Research*, Vol. 44, No. 22, pp.4855–4875.

Elbenani, B. and Ferland, J.A., (2012) ‘Cell formation problem solved exactly with the dinkelbach algorithm’, *CIRRELT*, pp.1-14.

Ghosh, T. and Dan, P.K., (2012) ‘An effective machine-part grouping algorithm to construct manufacturing cells’, Vol. 1, pp.1-5.

Ghosh, T., Sengupta, S., Chattopadhyay, M., and Dan, P., (2011) ‘Meta-heuristics in cellular manufacturing: A state-of-the-art review’, *International Journal of Industrial Engineering Computations,* Vol. 2, No. 1, pp.87-122.

Gonçalves, J.F. and Resende, M.G., 2004. An evolutionary algorithm for manufacturing cell formation. *Computers & industrial engineering,* Vol. 47, No. 2-3, pp.247-273.

Hazarika, M. and Laha, D., (2015) ‘Machine-part cell formation for maximum grouping efficacy based on genetic algorithm’, *In 2015 IEEE Workshop on Computational Intelligence*, pp.1-6.

Hazarika, M. and Laha, D., (2017) ‘Application of genetic algorithm in generalized machine cell formation problem’, *In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) IEEE,* pp.1855-1860.

Hosseinabad ER, and Zaman, MAU, (2020) ‘A Brief Review on Cellular Manufacturing and Group Technology”, *Research Journal of Management Reviews*, Vol. 5, no. 1, pp. 1-20.

Iqbal, M.A., Khan, N.K., Jaffar, M.A., Ramzan, M. and Baig, A.R., (2010) ‘April. Opposition based genetic algorithm with Cauchy mutation for function optimization’, *In 2010 International Conference on Information Science and Applications*, pp.1-7.

Izquierdo, R. C., Lorini, F. J., & Gomes, H. M. (2016) ‘Comparative analysis between efficiency grouping and efficacy grouping in cell formation using the firefly metaheuristic algorithm’, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Vol. 230, No. 8, pp.1548–1558.

Javaid, W., Tariq, A. and Hussain, I., (2014) ‘A comparison of a Standard Genetic Algorithm with a hybrid genetic algorithm applied to cell Formation Problem’, *Advances in Mechanical Engineering*, pp.1-11. http://dx.doi.org/10.1155/2014/301751.

Liu, C., Yasuda, K., Yin, Y., & Tanaka, K. (2008) ‘Uncertain association rule mining algorithm for the cell formation problem in cellular manufacturing systems’, *International Journal of Production Research*, Vol. 47, No. 3, pp.667–685.

Mahdavi, I., Paydar, M.M., Solimanpur, M. and Heidarzade, A., (2009) ‘Genetic algorithm approach for solving a cell formation problem in cellular manufacturing’, *Expert Systems with Applications*, Vol. 36, No. 3, pp.6598-6604.

Manimaran, A., Nagaraj, G., Venkumar, P. and Ganesamoorthy, R., (2014) “Manufacturing cell formation considering various production factors using modified art1 networks,” *Academic Journal of Manufacturing Engineering*, Vol. 12, No.3/2014, pp 55 – 61.

Manimaran, A., Nagaraj, G., Venkumar, P. and Paramasamy, S., (2013) ‘Manufacturing cell formation using back propagation networks’, *Elixir Mech. Engg*. Vol. 62, pp. 17803-17808.

Mukattash, A., Dahmani, N., Al-Bashir, A. and Qamar, A., (2018) ‘Comprehensive grouping efficacy: A new measure for evaluating block-diagonal forms in group technology’, *International Journal of Industrial Engineering Computations*, Vol. 9, No. 1, pp.155-172.

Nagaraj, G., Arunachalam, M., Vinayagar, K. and S. Paramasamy, (2020) ‘Enhancing performance of cell formation problem using hybrid efficient swarm optimization’, *Soft Computing*, https://doi.org/10.1007/s00500-020-05059-4.

Nagaraj, G., Manimaran, A., Rakesh, A.S., Sheik Syed Abuthahir S. and Paramasamy, S., (2015) “Comparison of Matrix Clustering Methods to Design Cell Formation”, *International Journal of Applied Engineering Research*, ISSN 0973-4562 Vol. 10, No.28, pp.21900– 21904.

Nagaraj, G., Manimaran, A., Rakesh, A.S., Sheik Syed Abuthahir S. and Paramasamy, S., (2015) “Productivity Enhancement by Balancing Takt Time in a Cellular Manufacturing Industry”, *International Journal of Applied Engineering Research*, ISSN 0973-4562 Vol. 10, No.28, pp. 22040 – 22044.

Neufeld, J. S., Teucher, F. F., & Buscher, U. (2019) ‘Scheduling flowline manufacturing cells with inter-cellular moves: non-permutation schedules and material flows in the cell scheduling problem’, *International Journal of Production Research*, pp.1–17.

Noktehdan, A., Karimi, B., & Husseinzadeh Kashan, A. (2010) ‘A differential evolution algorithm for the manufacturing cell formation problem using group-based operators’, *Expert Systems with Applications*, Vol. 37, No. 7, pp.4822–4829.

Noktehdan, A., Seyedhosseini, S. and Saidi-Mehrabad, M., (2016) ‘A Metaheuristic algorithm for the manufacturing cell formation problem based on grouping efficacy’, *The International Journal of Advanced Manufacturing Technology*, Vol. 82, No. 1-4, pp.25-37.

Paramasamy, S., Manimaran, A., Vinayagar, K. and Nagaraj, G., (2019) ‘Cell formation in sheet metal processing industry using genetic algorithm’, *Caribbean Journal of Science*, Vol. 53, No. 2, pp.2526-2532

Rabbani, M, Farrokhi-Asl, H & Ravanbakhsh, M, (2019), ‘Dynamic cellular manufacturing system considering machine failure and workload balance’, *Journal of Industrial Engineering International*, Vol. 15, no. 1, pp. 25-40.

Rajesh, K.D., Chalapathi, P.V., Chaitanya, A.B.K., Sairam, V. and Anildeep, N., (2006) ‘Modified grouping efficacy and new average measure of flexibility: performance measuring parameters for cell formation applications’, *ARPN Journal of Engineering and Applied Sciences*, Vol. 11, No. 15, pp.9212-9215.

Selim, H. M. (2002) ‘Manufacturing cell formation problem: a graph partitioning approach’, *Industrial Management & Data Systems*, Vol. 102, No. 6, pp.341–352.

Tunnukij, T. and Hicks, C., (2009) ‘An enhanced grouping genetic algorithm for solving the cell formation problem’, *International Journal of Production Research*, Vol. 47, No. 7, pp.1989-2007.

Ulutas, B.H., (2018) 'An immune system based algorithm for cell formation problem', *Journal of Intelligent Manufacturing*, pp.1-18. https://doi.org/10.1007/s10845-018-1407-x

Wang, H., Li, C., Liu, Y. and Zeng, S., (2007) 'A hybrid particle swarm algorithm with Cauchy mutation', *In 2007 IEEE Swarm Intelligence Symposium*, pp. 356-360.

Wu, Q., & Law, R. (2011) 'Cauchy mutation based on objective variable of Gaussian particle swarm optimization for parameters selection of SVM', *Expert Systems with Applications,* Vol. 38, No. 6, pp.6405–6411.

Xambre, A.R. and Vilarinho, P.M., (2003) 'A simulated annealing approach for manufacturing cell formation with multiple identical machines', *European journal of operational research,* Vol. 151, No. 2, pp.434-446.

Yang, Y., Chen, Y., & Long, C. (2016). Flexible robotic manufacturing cell scheduling problem with multiple robots. *International Journal of Production Research*, Vol. 54, No. 22, pp.6768–6781.

Zhao, C., & Wu, Z. (2000) 'A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives', *International Journal of Production Research,* Vol. 38, No. 2, pp.385–395.

Zhou, G., Zhang, C., Li, Z., Ding, K., & Wang, C. (2020) 'Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing', *International Journal of Production Research*, Vol. 58, No. 4, pp.1034–1051.

Zohrevand, A.M., Rafiei, H. and Zohrevand, A.H., (2016) 'Multi-objective dynamic cell formation problem: A stochastic programming approach', *Computers & Industrial Engineering,* Vol. 98, pp.323-332.