Scientia Iranica (2024) 31(11), 888–905

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
https://scientiairanica.sharif.edu

Integrated cell formation and part scheduling: A new mathematical model along with two meta-heuristics and a case study for truck industry

Y. Buruk Sahin* **and S. Alpay**

Department of Industrial Engineering, Eskisehir Osmangazi University, Eskisehir, Turkey.

Received 8 September 2021; received in revised form 30 October 2022; accepted 30 January 2023

KEYWORDS
Cell formation (CF);
Marine Predators Algorithm (MPA);
Genetic Algorithm (GA);
Exceptional elements;
Make-span;
Voids.

Abstract. This paper proposes a new linearized mathematical model to solve the integrated cell formation and job scheduling problem. The model aims to minimize the exceptional elements, voids, and the make-span of the jobs. The results of test problems show that the proposed model is very effective to obtain the best solutions for small-sized problems in reasonable computation times. However, due to the NP-hard nature of the considered problem, the best solutions couldn't be obtained in acceptable times for large-sized test problems, whereas the real-life applications of the problem addressed here are often much larger in size. To meet the requirement of solving larger-sized problems, the Genetic Algorithm (GA), which is today considered one of the artificial intelligence and machine learning techniques, and the Marine Predators Algorithm (MPA) as a new and nature-inspired metaheuristic, are proposed. The success of the algorithms was investigated and compared. The test results reveal the fact that the MPA with optimized parameters has a high potential to solve real life problems. At last, an attempt is made to re-design an existing real-life production system with the proposed algorithms. Eventually, a considerable improvement is obtained in performance compared to the current situation of the system.

1. Introduction

The Cellular Manufacturing System (CMS) is an effective production system that enables a group of machines to be grouped into a machine cell to process a group of products. CMS enables production control to be performed independently in each cell, with the aim of creating units that deal with a limited number of parts. CMS has many benefits, such as minimizing setup time, throughput time, material flow and transport times, work-in-process inventories, and finished product stocks [1].

CMS includes the following decisions: "Cell Formation (CF)", "Group Layout (GL)", "Cellular Part Scheduling (CPS)," and "Resource Allocation (RA)". The formation of cells is known as the principal activity

*. *Corresponding author. Tel.: +90 222 239 37 50–3610; Fax: +90 0 222 239–3613*
E-mail addresses: yelizburuk@ogu.edu.tr (Y. Buruk Sahin); salpay@ogu.edu.tr (S. Alpay)

To cite this article:
Y. Buruk Sahin and S. Alpay "Integrated cell formation and part scheduling: A new mathematical model along with two meta-heuristics and a case study for truck industry", *Scientia Iranica*, (2024) **31**(11), pp. 888-905.
DOI: 10.24200/sci.2023.59026.6023

in cell design. This phase includes the assignment of parts families and machines to the cells to be formed [2]. However, some machines required for some operations for the part families on certain cells may be unavailable. Therefore, it may be necessary to process some parts in machines in two or more cells. The requirement arising from that intercellular movement of a part between cells is called Exceptional Element (EE) [3]. CMS designers try to minimize the number of EEs to avoid intercellular movements and, in turn, to obtain shorter completion times and, consequently, lower costs. Customarily, CMS designers were satisfied with part-machine grouping only by ignoring job scheduling. However, it is more realistic to consider the CF problem together with CPS.

Scheduling is a decision-making process encountered in many manufacturing and service industries. In this process, allocating resources to jobs and sequencing them in certain periods is aimed to optimize one or more objectives [4]. Li et al. [5] note that: "The objective of CPS is to identify the sequence of parts in cells that minimizes some effectiveness criteria".

In this study, Flexible Jop Shop Scheduling (FJSS) is considered. Amongst the scheduling, make-span is selected to utilize the advantage of idle time reduction to improve machine utilization. Moreover, minimizing make-span will directly minimize the production costs [6]. In FJSS, each job operation can be processed on any machine that is previously selected among a set of machines [7].

Compared to FJSS literature, there are relatively few publications dealing with CF and CPS together in CMS. Wu et al. [8] considered CF, machine layout, and scheduling decisions in CMS with the aim of minimizing make-span and proposed a hierarchical Genetic Algorithm (GA) to solve the integrated cell design problem. The results show that the simultaneous solution process is more successful than the sequential one. Wang et al. [9] aimed to minimize the total delay by using a penalty cost in an integrated decision process of CPS. Ghezavati and Saidi-Mehrabad [10] considered a stochastic environment where the processing times of jobs in different machines are random. CF and CPS decisions are handled concurrently by a hybrid method. Kesen et al. [11] developed a heuristic approach based on GA for job scheduling problems in virtual cells where duplicate machines and alternative sequences and routes are assumed. Tang et al. [12] considered the problem of scheduling parts in CMS sequentially with the aim of minimizing intra-cell movements and proposed a random search algorithm. Aryanezhad et al. [13] considered assembly processes in CMS. Li et al. [5] considered the issue of scheduling parts in CMS. Ant Colony Optimization approach (ACO) is used on flexible routes. Tang et al. [14] presented an integrated approach for CPS decisions, taking into consideration CF and part scheduling by mathematical optimization and random search approaches. Liu and Wang [15] proposed a non-linear integer mathematical model to solve the CF and scheduling problem in a dual-resource-constrained setting. Rafiei et al. [16] considered the problem of CPS and proposed a non-linear mixed-integer model and a hybrid approach. Liu et al. [17] researched the problem of CF and job scheduling in a multi-functional worker and machine environment by proposing a Discrete Bacteria Foraging Algorithm (DBFA). Iqbal et al. [18] addressed the CF and CPS to ensure minimum energy consumption and make-span. Costa et al. [19] studied a CMS and investigated the flow-shop sequence-dependent group scheduling problem with the objective of make-span minimization. They propose a hybrid metaheuristic method that integrates GA and random sampling search methods. Buruk Sahin and Alpay [3] developed a new mathematical model for the optimization of CPS and CF problems. They aim to minimize the EEs in cells and the make-span of the jobs, simultaneously. Feng et al. [20] showed that the integration of CF and CPS can remarkably reduce the flow time of CMS. Far et al. [21] dealt with the CPS problem in deterministic and fuzzy environments. Forghani and Ghomi [22] presented classical and virtual cell configurations. They integrated CF, GL, and CPS problems with the aim of minimizing total handling costs and average cycle time.

Some of the design attributes of the models considered in CF and CPS literature are intercellular transportation times [14,15,23], cell size restriction [15], operation sequence and processing times [14,15,23,24], duplicate machines [20,25], setup times [16,19,23], reentrant parts [20,26], energy efficient routings [18], non-permutation schedules [27], variable cell number [20], maintenance [28], learning and forgetting effects [29,30], fuzzy model [31], and capacity measurement [32].

Some recent papers integrating CF and CPS decisions in CMS, along with the objectives of the models, the solution approaches, and the software used, are summarized in Table 1. Due to the critical role in determining the performance of the algorithms, the parameter tuning approaches in the studies are also examined.

Although there are studies in which CF and CPS are taken independently of each other in the relevant literature, many studies have shown that the CF problem and the CPS problem are interrelated and should not be considered separately [8,18,20,24,26]. Feng et al. [20] emphasize that integrated CF and scheduling decisions are correlated, and the integrated approach may provide time savings of up to 13.2% by offering more successful solutions than the sequential approach.

The main contributions of this study are explained as follows:

Table 1. Literature review of part scheduling in Cellular Manufacturing System (CMS).

No.	Reference	Problem	Objective (s)	Solution method (s)	Software	Parameter tuning
1	Tang et al. [14]	Jop shop	Tardiness penalty cost	Mathematical model Lagrangian relaxation decomposition with a heuristic	Gams Java	–
2	Halat and Bashirzadeh [23]	Jop shop	Makespan	Mathematical model heuristic-based on GA	Lingo MATLAB	Factorial design
3	Liu and Wang [15]	Jop shop	Makespan	Mathematical model hybrid SA algorithm	C++	–
4	Rafiei et al. [16]	Jop shop	Intercell/intracell transportation costs makespan	Hybrid SA/GA	Gams MATLAB	–
5	Liu et al. [17]	Jop shop	Material handling costs fixed and operating costs of machines/workers	Discrete bacteria foraging algorithm	C++	–
6	Costa et al. [19]	Flow shop	Makespan	Hybrid metaheuristic	MATLAB	ANOVA
7	Buruk Sahin and Alpay [3]	Jop shop	Makespan EE	Mathematical model	Gams	–
8	Feng et al. [20]	Jop shop	Flowtime	Improved GA	Lingo	–
9	Neufeld et al. [27]	Flowline	Total makespan	SA algorithms	C	Full factorial design
10	This study	Jop shop	Makespan EE void	Mathematical model GA MPA	Gams MATLAB	Taguchi

- Different from the others on CF and parts scheduling problems, a new mathematical model has been developed for the first time considering voids, EEs, and make-span simultaneously;
- A GA and a Marine Predators Algorithm (MPA) are proposed for large-sized instances of the addressed problem to produce good/applicable solutions in reasonable times;
- The proposed algorithms are applied to a real-life problem, and considerable improvements are observed compared to present practices.

2. Problem formulation and mathematical model

In this study, a new mathematical model with the following assumptions is developed. Parts can be processed by different types of machines. In each type, there exists one machine. Each machine and each part can be allocated to only one cell. Some part operations can be processed on an alternative machine set. In this set, the processing times can be the same or different from each other. Each part has a sequence of operations processed in a given order. The processing time for the operations of each part type on a machine is known and fixed. No preemption is allowed. Set-up times for the parts are assumed sequence-independent and considered to be included in the processing times. The ith operation of each job can only be started after the $(i-1)$th operation of that part has been completed and the time at which the required machine has completed its operation of the current job. There is a lower bound for the number of machines and parts allocated to each cell.

Indices

- i Index for the part types $i = \{1, \ldots, n\}$
- j Index for the operations required by parts $j = \{0, \ldots, n_i\}$
- k Index for the machine types $k = \{0, \ldots, m\}$
- l Index for the position number of the assigned operation in the machine $l = \{0, \ldots, l_i\}$
- c Index for the cells

Parameters

- n Total number of parts
- m Total number of machines
- j_i Total number of operations of job i
- O_{ij} ith operation of part j
- p_{ijk} The standard time to process O_{ij}
- a_{ijk} 1 if operation O_{ij} is required processing on machine k, and 0 otherwise
- M A large positive number
- w_p Weight of objective p

Decision variables

- $x_{i,j,k,l,c}$ 1 if O_{ij} is processed on machine k in the lth position in cell c, and 0 other wise
- $z_{k,c}$ 1 if machine type k is assigned to cell c, and 0 other wise
- $y_{i,c}$ 1 if part type i is assigned to cell c, and 0 other wise
- $v_{i,k,c}$ 1 if part i is processed on machine k in cell c, and 0 other wise
- $t_{i,j}$ Starting time of O_{ij}
- $Tm_{k,l}$ Starting time of O_{ij} (in machine k and in the lth order)
- EE Exceptional Element
- $C_{\max}$ Maximum completion time
- Z_{obj} Scalarized objective function

Objective functions and constraints

$$\text{Min } Z_{obj} = \sum_i \sum_j \sum_k \sum_l \sum_c x_{i,j,k,l,c} * z_{k,c} * (1 - y_{i,c}) + \sum_i \sum_k \sum_c y_{i,c} * z_{k,c} * (1 - v_{i,k,c}) + C_{\max}, \quad (1)$$

$$\sum_k \sum_l \sum_c x_{i,j,k,l,c} = 1 \qquad \forall i, j, \quad (2)$$

$$\sum_i \sum_j x_{i,j,k,l,c} \leq 1 \qquad \forall k, l, c, \quad (3)$$

$$\sum_l \sum_c x_{i,j,k,l,c} \leq a_{i,j,k} \qquad \forall i, j, k, \quad (4)$$

$$t_{i,j+1} \geq t_{i,j} + \sum_k \sum_l x_{i,j,k,l,c} * p_{i,j,k} \qquad \forall i, j < n_j, c, \quad (5)$$

$$Tm_{k,l+1} \geq Tm_{k,l} + \sum_i \sum_j x_{i,j,k,l,c} * p_{i,j,k} \qquad \forall k, l < l_i, c, \quad (6)$$

$$Tm_{k,l} \leq t_{i,j} + M * (1 - x_{i,j,k,l,c}) \qquad \forall i, j, k, l, c, \quad (7)$$

$$Tm_{k,l} \geq t_{i,j} - M * (1 - x_{i,j,k,l,c}) \qquad \forall i, j, k, l, c, \quad (8)$$

$$C_{\max} \geq t_{i,j} + \sum_k \sum_l x_{i,j,k,l,c} * p_{i,j,k} \qquad \forall i, \quad (9)$$

$$\sum_c z_{k,c} = 1 \qquad \forall k, \quad (10)$$

$$\sum_k z_{k,c} \geq 1 \qquad \forall c, \quad (11)$$

$$\sum_c y_{i,c} = 1 \qquad \forall i, \quad (12)$$

$$\sum_i y_{i,c} \geq 1 \qquad \forall c, \quad (13)$$

$$x_{i,j,k,l,c} \leq a_{i,j,k} * z_{k,c} \qquad \forall i, j, k, l, c, \quad (14)$$

$$z_{k,c}, y_{i,c}, x_{i,j,k,l,c} \in \{0,1\} \qquad \forall k,c \quad \forall i,c \quad \forall i,j,k,l,c, \tag{15}$$

$$Tm_{k,l}, t_{i,j}, C_{\max}, Z_{obj} \geq 0 \qquad \forall k,l \quad \forall i,j. \tag{16}$$

In the model, Eq. (1) shows the scalarized objective function. It includes three sub-objectives: minimization of EE, minimization of number of voids, and minimization of make-span. Constraint (2) imposes that each part process is assigned to one cell and only one position of the existing machines. Constraint (3) states that some capabilities of some machines cannot be fully used depending on alternative machine sets. Constraint (4) guarantees that each operation is performed on the respective predetermined machines. Constraint (5) ensures the precedence relationships related to the starting time of the operations. Constraint (6) ensures that, in assignments for each position of a machine, the occupancy of previous positions of that machine is controlled. Constraints (7) and (8) provide that an operation can be assigned to only one position of a machine. Constraint (9) determines the maximum completion time of operations on all available machines by considering the last completed time for all operations. Constraint (10) ensures that each machine can be assigned to one cell. Constraint (11) ensures a lower bound for assigning machines to cells. Constraint (12) determines that only one cell must be assigned for each part. Constraint (13) ensures a lower bound for assigning parts to cells. Constraint (14) guarantees that each part operation can be done only in the cell to which a relevant machine is assigned. Constraints (15) and (16) illustrate the binary and continuous decision variables, respectively.

The proposed model is non-linear, and auxiliary binary variables ($F_{i,j,k,l,c}$, $S_{i,j,k,l,c}$, $U_{i,k,c}$, $H_{i,k,c}$) were used to reformulate the model by introducing new sets of variables. The variables, $F_{i,j,k,l,c}= x_{i,j,k,l,c}* z_{k,c}$, $S_{i,j,k,l,c} = x_{i,j,k,l,c} *z_{k,c}* y_{i,c}$, $U_{i,k,c} = y_{i,c}* z_{k,c}$, and $H_{i,k,c}= y_{i,c}* z_{k,c}* v_{i,k,c}$ were used to linearize terms in the objective function. Additionally, $v_{i,k,c}$ is determined to calculate the number of voids. Constraints (17)–(28) were used to linearize the model. Eq. (29) shows the scalarized weighted objective function in the linear structure.

$$x_{i,j,k,l,c} + z_{k,c} \geq 2 * F_{i,j,k,l,c} \qquad \forall i,j,k,l,c, \tag{17}$$

$$x_{i,j,k,l,c} + z_{k,c} \leq 1 + F_{i,j,k,l,c} \qquad \forall i,j,k,l,c, \tag{18}$$

$$x_{i,j,k,l,c} + z_{k,c} + y_{i,c} \geq 3 * S_{i,j,k,l,c} \qquad \forall i,j,k,l,c, \tag{19}$$

$$x_{i,j,k,l,c} + z_{k,c} + y_{i,c} \leq 2 + S_{i,j,k,l,c} \qquad \forall i,j,k,l,c, \tag{20}$$

$$y_{i,c} + z_{k,c} \geq 2 * U_{i,k,c} \qquad \forall i,k,c, \tag{21}$$

$$y_{i,c} + z_{k,c} \leq 1 + U_{i,k,c} \qquad \forall i,k,c, \tag{22}$$

$$y_{i,c} + z_{k,c} + v_{i,k,c} \geq 3 * H_{i,k,c} \qquad \forall i,k,c, \tag{23}$$

$$y_{i,c} + z_{k,c} + v_{i,k,c} \leq 2 + H_{i,k,c} \qquad \forall i,k,c, \tag{24}$$

$$v_{i,k,c} \leq \sum_j \sum_l x_{i,j,k,l,c} \qquad \forall i,k,c, \tag{25}$$

$$F_{i,j,k,l,c}, S_{i,j,k,l,c}, U_{i,k,c}, H_{i,k,c}, v_{i,k,c} \in \{0,1\} \qquad \forall i,j,k,l,c, \tag{26}$$

$$EE = \sum_i \sum_j \sum_k \sum_l \sum_c (F_{i,j,k,l,c} - S_{i,j,k,l,c}), \tag{27}$$

$$void = \sum_i \sum_k \sum_c \left(U_{i,k,c} - H_{i,k,c}\right), \tag{28}$$

$$\mathrm{Min} Z_{obj} = w_1 EE + w_2 void + w_3 C_{\max}, \tag{29}$$

$$EE, void, Z_{obj} \geq 0 \qquad \forall k, l \forall i, j. \tag{30}$$

Optimal results could not be reached for medium and large-sized test problems within reasonable times because of their combinatorial structures. As Chaudhry and Khan [7] note: "In terms of computational complexity, JSS problem is NP-hard. So, for even small instances, an optimal solution cannot be guaranteed. Additionally, FJSS problem is more complex than JSSP as it considers the determination of machine assignment for each operation". On the other hand, most real-life CF and job scheduling problems are both larger in size and more complex in structure. As the problem size grows, the time required to reach the best solutions is far from being acceptable. Whereas time is quite valuable in competitive conditions of the practical business environment.

3. Solution methodology

To reach acceptable solutions in shorter/acceptable times in real-life problems, a GA and MPA are presented in the following section.

3.1. Genetic Algorithm (GA)

GA was first introduced by Holland [33], and today, it is considered one of the artificial intelligence and machine learning algorithms [34]. The structure of potential solutions to a problem is designed at the initial step of GA to constitute chromosomes. Each component of this chromosome is referred to as a gene, and a set of chromosomes is referred to as a population. An initial population consisting of feasible solutions is created randomly. GA includes "selection", "reproduction (crossover)," and "mutation" mechanisms. In the selection step, the chromosomes are elected by using a kind of biased random process from the population. Crossover enables the production of new feasible solutions, and mutation is used to increase the variety of the population. A new generation is formed by some of the parents. After several generations, GA converges to hopefully an optimal or suboptimal solution to the problem represented by the best chromosome of the last population. Fitness function is the measure of a chromosome's performance. A fitness function is proposed by this study as in Eq. (31):

$$\begin{aligned} Fitnessfunction = {} & total\ no.\ of\ exceptional\ elements \\ & + total\ no.of\ voids + make \\ & - span + total\ penalty. \end{aligned} \tag{31}$$

The total number of EEs is calculated by considering the total number of jobs that could not be performed in the assigned cell. A void refers to a part operation that does not require processing on a machine inside its own cell. Make-span has been calculated with the help of machine part operation matrices and by selecting the largest one among the completion times of the last jobs considering machine and part suitability times.

A penalty function is added to the fitness value for each chromosome in the population that violates any constraint in the mathematical model to eliminate unfeasible solutions [2,6]. The Total Penalty (TP) proposed by this study to eliminate the chromosomes that do not comply with Constraints (11) and (13) in the mathematical model and the infeasibility of the schedule are given in Eq. (32):

$$\begin{aligned} TP = {} & \varepsilon_1 * total\ (part\ related\ penalties) \\ & + \varepsilon_2 * total\ (machine\ related\ penalties) \\ & + \varepsilon_3 * total(time\ related\ penalties). \end{aligned} \tag{32}$$

In case of not assigning at least one piece, the penalty for the part is applied. A machine penalty is applied for each case in which at least one machine is not assigned to each cell. Non-feasible solutions regarding the calculation of the completion time are also reflected in the penalty function as time-related penalties, and the acceptance of these solutions is prevented. Other constraints in the model used in defining the problem are provided by the developed chromosome structure. The coefficients ε_1, ε_2, and ε_3 of the penalty function are adopted in accordance with the problem size and the magnitude of the objective function value.

```
Procedure GA:
Begin
  t ← 0;
  initialize P(t);
  evaluate P(t);
  while (not termination condition) do
     recombine P(t) to yield C(t);
     evaluate C(t);
     select P(t+1) from P(t) to C(t);
     t ← t+1;
  end
end
```

Figure 1. Pseudocode for Genetic Algorithm (GA).

GA parameters are generally thought to be determined by the experimental analysis [35]. A concise pseudocode of the proposed GA for the studied integrated problem is seen in Figure 1 [36]. Here, P(t) and C(t) are parents and offspring in the current generation t. Recombination involves crossover and mutation to yield offspring.

3.1.1. *Chromosome structure*

Developing a chromosome structure is the first step in obtaining high-quality results for the problem. There are various chromosome structures for FJSS and CF problems [16]. Figure 2 shows an example representation of the designed chromosome structure for the considered problem in this study. It consists of three sections for assignments to cells, assignments to machines, and job-operation sequences.

The first section of the chromosome includes as many genes as the total number of machines and parts. The assignment of machines and parts to cells is represented in this section. The size of the second section is equal to the total number of part-operations assigned to the machines. Each value in this structure represents the chosen machine alternative for the related part-operation. The chromosome represents operations sequentially from left to right. The third

Assignments to cells								Assignments to machines operations						Job-operation sequence operations					
Machines					Parts														
M1	M2	M3	M4	..	P1	P2	..	O11	O12	O13	O21	O22	..	O11	O12	O13	O21	O22	..

Figure 2. Chromosome representation.

section that represents the operation sequence includes as many genes as the total number of part-operations and represents the operations sequentially from left to right. This section also represents the replacements of operations on the Gantt chart. Finally, chromosome size is represented by the following equation: number of parts + number of machines + 2*total number of operations.

3.1.2. Deciding on genetic operators

A number of methods, such as genetic operator alternatives, are found in the literature to construct a proper GA. Three types of "selection" methods, namely, "Tournament selection", "Roulette wheel selection," and "Stochastic uniform selection" [37], three types of "*crossover*" operators, namely, "One-point crossover", "Two-point crossover" and "Scattered crossover," and "adaptive feasible mutation" as the "*mutation*" operator [38] are considered in this study. It should also be noted that the chromosome structure proposed in the study prevents obtaining unfavorable solutions after crossover and mutation processes. An experimental design based on the Taguchi technique is used to decide on the mentioned operators and calculate the optimum values of the parameters [35].

3.2. Marine Predators Algorithm (MPA)

Faramarzi et al. [39] developed MPA inspired by the Lévy and Brownian motions in ocean predators. MPA is a meta-heuristic optimization algorithm that simulates the hunting process based on the relationship between prey and predators in the sea. While marine predators exhibit Brownian motion in half of this hunting period, they spend the remaining half in Lévy motion. Predators aim to maximize the possibility of catching their prey with such different movement patterns.

In the MPA, the best course of action for catching prey is of great importance, and the MPA tries to maintain a balance in the Lévy and Brownian motions. In this way, MPA provides an opportunity to evaluate different strategies for optimizing the hunting process [40].

For generation of the initial population, below Eq. (33) is used:

$$X_{ij} = bL_j + R \times (bU_j - bL_j) \qquad \begin{array}{l} i = 1, 2, \cdots, n \\ j = 1, 2, \cdots, d \end{array} \tag{33}$$

where n denotes the population size and d denotes the dimension of the search agent. R is the uniform random number vector, and bU_j and bL_j denote the upper and lower bounds, respectively, of the search agent in the jth dimensional search space.

Using Eq. (33), the Prey matrix is constructed, and the fitness values for all individuals are calculated. Then, the Elite matrix is formed from the ones called top predators with optimal fitness values of the same size as the Prey matrix. Elite matrix and top predators are updated with each iteration. For this reason, an individual who was a predator before may become the prey of other top predators later on. Also, the Prey matrix is updated depending on the different velocity ratios. Thus, a prey will be able to be in a different position in each iteration and get displaced. Below, Prey and Elite matrixes are expressed:

$$Elite = \begin{bmatrix} X^I_{1,1} & X^I_{1,2} & \cdots & X^I_{1,d} \\ X^I_{2,1} & X^I_{2,2} & \cdots & X^I_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X^I_{n,1} & X^I_{n,2} & \cdots & X^I_{n,d} \end{bmatrix}_{n \times d} \tag{34}$$

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ X_{3,1} & X_{3,2} & \cdots & X_{3,d} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n \times d} \tag{35}$$

Depending on the different movement rates of Predator and Prey, there are 3 movement phases in the MPA optimization process:

(a) High movement rate: At this phase, the prey has a higher movement rate than the predator. This phase is called the Exploration phase and is applied while the current iteration (*Iter*) $< \frac{1}{3}$ *IterMax* (maximum number of iterations). Required calculations are made with the following expressions:

$$\begin{array}{l} S_i = R_B \otimes (E_i - R_B \otimes X_i) \\ X_i = X_i + P \times R \otimes S_i \end{array} \qquad i = 1, 2, ..., n \tag{36}$$

where R denotes the uniform random number vector between 0 and 1, R_B denotes Brownian motion, $\otimes$ represents entry-wise multiplications, and P is a constant value of 0.5.

(b) Unit movement rate: At this phase, the moving speeds for both prey and predator are uniform. The population is divided into two parts: the first part (prey) fulfills exploitation by employing Lévy motion, and the second part (predator) fulfills exploration by employing Brownian motion during $\frac{1}{3}$ *IterMax* $<$ *Iter* $< \frac{2}{3}$ *IterMax.* For Lévy motion of prey, the below expressions are used for calculation:

$$S_i = R_L \otimes (E_i - R_L \otimes X_i) \qquad i = 1, 2, \cdots, n/2$$

$$X_i = X_i + P \times R \otimes S_i, \tag{37}$$

where RL denotes Lévy motion.

For the Brownian motion of the predator, the following expressions are used for calculation:

$$S_i = R_B \otimes (R_B \otimes E_i - X_i) \qquad i = n/2, \cdots, n$$

$$X_i = E_i + P \times CF \otimes S_i$$

$$CF = \left(1 - \frac{Iter}{Iter_{\max}}\right)^{\left(2\frac{Iter}{Iter_{\max}}\right)} \tag{38}$$

where CF denotes the step size control parameter for the predator movement.

(c) Low movement rate: The predator has a higher movement rate than the prey. This phase is called the Exploitation phase and is applied while the current iteration (*Iter*) > $\frac{2}{3}$ *IterMax*. The necessary calculations are made with the following expressions:

$$\begin{aligned} S_i &= R_L \otimes (R_L \otimes E_i - X_i) \qquad i = 1, 2, ..., n \\ X_i &= E_i + P \times CF \otimes S_i \end{aligned} \tag{39}$$

In the MPA process, it is also thought that environmental factors such as eddy formation or Fish Aggregating Devices (FADs) may have an impact on the behavior of marine predators, and this effect can be calculated with the help of the following expressions:

$$X_i = \begin{cases} X_i + CF\left[bL + R \otimes (bU - bL)\right] \otimes U \\ \qquad \text{if } r \leq FADs \\ X_i + \left[FADs\,(1 - r) + r\right](X_{r1} - X_{r2}) \\ \qquad \text{if } r > FADs \end{cases} \tag{40}$$

where U is a binary vector in which each array contains only 0 and 1, r denotes the uniform random number between 0 and 1, and subscripts $r1$ and $r2$ denote two randomly selected index values of the prey matrix.

Figure 3 shows the pseudocode associated with the working structure of MPA [41].

```
Initialize the prey population, i = 1,…,n
Iter = 1
while (Iter < Iter_max)
Calculate the fitness values of prey and establishment the elite matrix
if (Iter < Iter_max/3)
  Update the prey by Equation (36)
else if (Iter_max/3 < Iter < 2* Iter_max/3)
    if (i < n/2)
     Utilize Equation (37) to Update the prey
  else
   Use Equation (38) to Update the prey
else if (Iter > 2* Iter_max/3)
   Update the prey based on Equation (39)
    end if
    Accomplish the memory saving and update the elite matrix
    Execute the FADs effect by Equation (40)
    Iter++
end while
```

Figure 3. The Pseudo code belonging to the Marine Predators Algorithm (MPA).

In addition to the algorithm given in Figure 3, it should be noted that for solving the problem addressed in this study, the chromosome structure of MPA is also the same as for GA above.

4. Computational analysis

Since an original problem is considered, no test problem that matches all terms of the model is found in the literature. Nevertheless, the test problems given by Fattahi et al. [42] for FJSS are modified in this study to test the scheduling performance of the proposed model by adding some randomly produced parameter values. The data sets include 20 test problems. Three sets of test problems, small size (SFJSCF_1-10), and medium and large size (MFJSCF_1-20) are derived by means of (n, h, m, c) schema where n denotes the number of jobs, h denotes the number of operations, m denotes the number of machines, and c denotes the number of cells. Those test problems include the number of jobs vary 2 to 12, number of machines vary 2 to 8, and number of operations vary 4 to 48. All generated test problems structured for the proposed mathematical model are solved by GAMS 24.2.1 software CPLEX solver. The proposed GA and MPA are coded in MATLAB R2017b on a PC with an Intel CORE (Tm) i5-3330 CPU, 3.2 GHz processor, and 8 GB RAM to solve the test problems.

To improve the performance of GA, different experimental design techniques are used in the literature. Taguchi experimental design is one of the most effective techniques for convenient parameter settings in terms of more performance of GA [43–47]. Therefore, in this study, Taguchi's experimental design has been used to determine the proper types of operators and optimum levels of parameters of the proposed GA [48,49]. Table 2 shows the determined parameter levels of GA. When the parameters are evaluated in view of MPA, and the parameter value of P for MPA is 0.5. The population size and the maximum number of iterations $Iter_{\max}$ are set to 50 and 500, respectively.

Optimal solutions are obtained for small-sized test problems in acceptable computation times using the mathematical model as well as GA and MPA. Mathematical model performance on the small-sized test data is very effective, and the global optimal solutions could be obtained for problems of that size. Table 3 shows the computational results for both GA and MPA for small-sized test problems. In the table, Z_{GA} and Z_{MPA} denote the objective function (fitness) values of the GA and MPA, respectively. On the other hand, for medium- and large-sized problems, the computational results in terms of the best values that could be reached in 3600 seconds (1 hour) by GAMS are presented in Table 4. In the table, Z_{MM} denotes the corresponding bound value for related problems.

Table 2. Genetic Algorithm (GA) parameters and levels.

	Factors	Levels 1	Levels 2	Levels 3
A	Population size	30 (small) 2000 (large)	50 (small) 3000 (large)	–
B	Crossover operator	One-point	Two-point	Scattered
C	Mutation rate	0.05	0.10	0.15
D	Crossover rate	0.7	0.8	0.9
E	Selection operator	Roulette	Stochastic	Tournament

Table 3. Small-sized test problems and parameters.

Problems	Size (part, operation, machine, cell)	GA CPU	GA ($C_{\max}$, EE, $void$)	GA Z_{GA}	MPA CPU	MPA ($C_{\max}$, EE, $void$)	MPA Z_{MPA}	Mathematical model CPU	Mathematical model ($C_{\max}$, EE, $void$)	Mathematical model Z_{MM}	$\Delta_{GA}(\%)$ and $\Delta_{MPA}(\%)$
SFJSCF_1	(2,2,2,2)	2.5	(66,0,0)	66	0.063	(66,0,0)	66	0.25	(66,0,0)	66	0.0
SFJSCF_2	(2,2,2,2)	2.1	(107,0,0)	107	0.175	(107,0,0)	107	0.09	(107,0,0)	107	0.0
SFJSCF_3	(3,2,2,2)	2.9	(221,1,0)	222	0.200	(221,1,0)	222	4.4	(221,1,0)	222	0.0
SFJSCF_4	(3,2,2,2)	2.3	(355,1,0)	356	0.100	(355,1,0)	356	3.4	(355,1,0)	356	0.0
SFJSCF_5	(3,2,2,2)	2.8	(119, 3,0)	122	0.103	(119, 3,0)	122	16.0	(119,3,0)	122	0.0
SFJSCF_6	(3,3,3,2)	2.5	(320,0,0)	320	0.343	(320,0,0)	320	56.0	(320,0,0)	320	0.0
SFJSCF_7	(3,3,5,2)	3.6	(397,1,0)	398	1.125	(397,1,0)	398	5.0	(397,1,0)	398	0.0
SFJSCF_8	(3,3,4,2)	3.4	(253,1,0)	254	0.375	(253,1,0)	254	505	(253,1,0)	254	0.0
SFJSCF_9	(3,3,3,2)	4.9	(210,2,0)	212	0.360	(210,2,0)	212	45.2	(210,2,0)	212	0.0
SFJSCF_10	(4,3,5,2)	4.0	(516,0,0)	516	0.750	(516,0,0)	516	195	(516,0,0)	516	0.0

Table 4 also shows the best integer values obtained by GA and MPA in terms of fitness values and CPU times. Please note that considering the numbers of parts, machines, and cells in the relevant literature, the number of cells is assumed to be "2" in small size problems, "2" and "3" in medium and large sizes. $\Delta_{GA}(\%)$ and $\Delta_{MPA}(\%)$ express the variations between the results of the developed mathematical model and the proposed algorithms GA and MPA, respectively, and are calculated as follows:

$$\Delta_{GA}(\%) = 100x\frac{Z_{MM} - Z_{GA}}{Z_{MM}}, \tag{41}$$

$$\Delta_{MPA}(\%) = 100x\frac{Z_{MM} - Z_{MPA}}{Z_{MM}}. \tag{42}$$

From Table 4, one can see that the proposed GA and MPA are very effective in obtaining acceptable and reasonable solutions for medium- and large-sized problems in much shorter computational times compared to results obtained by the developed model. MPA can also achieve much better fitness values as size increases. When considering the NP nature of the problem and the dimensions and complexity of real-life problems are much higher than the test problems, it is certain that the potential of the MPA algorithm to produce workable good solutions will gain more importance.

5. A real-life application

The proposed GA and MPA algorithms have been applied to a large-scale real-life problem at the gear-cutting shop of a world-class truck manufacturer to test their performances by comparing them with present

Table 4. Medium/large sized test problems and parameters.

Problems	Size (part, operation, machine, cell)	GA				Marine Predators Algorithm				Mathematical model		
		CPU (s)	($C_{\max}$,EE, void)	Z_{GA}	Δ_{GA} (%)	CPU (s)	($C_{\max}$,EE, void)	Z_{MPA}	Δ_{MPA} (%)	CPU (s)	($C_{\max}$,EE, void)	Z_{MM}
MFJSCF_1	(5,3,6,2)	41	(469,2,2)	473	0	1.78	(469,2,2)	473	0.0	3600	(469,2,2)	473
	(5,3,6,3)	94	(468,3,0)	471	0	2.40	(468,3,0)	471	0.0	3600	(468,3,0)	471
MFJSCF_2	(5,3,7,2)	73	(446,2,6)	454	0	1.90	(446,2,6)	454	0.0	3600	(446,2,6)	454
	(5,3,7,3)	97	(446,4,0)	450	+1	2.38	(446,4,0)	450	+1.1	3600	(448,5,2)	455
MFJSCF_3	(6,3,7,2)	247	(466,3,8)	477	0	4.07	(466,3,8)	477	0.0	3600	(466,3,8)	477
	(6,3,7,3)	143	(468,6,2)	476	+0.4	5.01	(468,6,2)	476	+0.4	3600	(468,7,3)	478
MFJSCF_4	(7,3,7,2)	178	(564,2,9)	574	0	11.5	(564,2,9)	574	0.0	3600	(565,1,8)	574
	(7,3,7,3)	184	(564,5,3)	572	+0.5	10.7	(564,5,3)	572	+0.5	3600	(565,5,5)	575
MFJSCF_5	(7,3,7,2)	192	(514,3,9)	526	0	12.3	(514,3,9)	526	0.0	3600	(514,3,9)	526
	(7,3,7,3)	135	(519,5,3)	527	+0.4	9.37	(514,8,5)	527	+0.4	3600	(514,9,6)	529
MFJSCF_6	(8,3,7,2)	162	(649,4,8)	661	+0.9	23.9	(641,9,6)	656	+1.6	3600	(648,15,4)	667
	(8,3,7,3)	202	(634,6,4)	644	+0.3	21.9	(634,6,4)	644	+0.3	3600	(634,10,2)	646
MFJSCF_7	(8,4,7,2)	214	(894,8,7)	909	0	91.6	(881,18,7)	906	+0.3	3600	(881,23,5)	909
	(8,4,7,3)	204	(910,13,5)	928	+1.3	83.7	(897,19,5)	921	+2.0	3600	(920,17,3)	940
MFJSCF_8	(9,4,8,2)	311	(944,6,11)	961	+2.5	262	(921,22,9)	952	+3.4	3600	(958,16,11)	985
	(9,4,8,3)	216	(925,13,4)	942	+5.1	171	(911,17,8)	936	+5.5	3600	(959,26,5)	990
MFJSCF_9	(11,4,8,2)	308	(1165,11,15)	1191	+0.3	432	(1150,15,8)	1173	+1.8	3600	(1158,22,15)	1195
	(11,4,8,3)	225	(1272,14,4)	1290	+6.0	318	(1163,23,11)	1197	+12.8	3600	(1346,20,7)	1373
MFJSCF_10	(12,4,8,2)	407	(1284,13,18)	1315	+5.8	497	(1294,10,12)	1316	+5.4	3600	(1361,18,12)	1391
	(12,4,8,3)	359	(1372,21,9)	1402	–	382	(1327,25,13)	1365	–	3600	–	–

practices. To produce different diameters and pitches of gears, there are 24 machines positioned by their functions. A total of 79 operations on 12 parts are processed in that system (see Appendix A). An integrated feasible solution to the cell forming and part scheduling problem could not be obtained even after running the mathematical model for 10 hours.

To provide more convenience and flexibility to decision makers, different weights in the objective function were used. Considering that large makespan values have more effect on the objective function, weight values for w_{EE} and w_{void} were taken as "1" and "4" different weights were investigated for $w_{c\max}$ as 1, 1/2, 1/4, and 1/8. The number of cells to be created for the problem had been selected as 3. The parameters and levels for the proposed GA and MPA were designed to be the same as in the previous section.

In the current situation, the cellular production

Table 5. Results for real-life application.

Real case	**Company's current heuristic method**	**GA**					**MPA**			
(w_{HDE}, w_{HB}, w_{Cenb})	**Objective function value**	**Best GA levels**	**Objectives (*EE*, *void*, $C_{\max}$)**	**Avg. GA**	**Best GA (% improvement rate)**	**CPU time (s)**	**Objectives (*EE*, *void*, $C_{\max}$)**	**Avg. MPA**	**Best MPA (% improvement rate)**	**CPU time (s)**
(1, 1, 1)	403	$A_2B_3C_1D_1E_2$	(43,20,165)	235.3	228 (43.4)	652	(24,46,144)	217,4	214 (46.90)	402
(1, 1, 1/2)	308.50	$A_2B_3C_3D_2E_2$	(35,19,172)	142.5	140 (54.62)	1094	(21,37,162)	140.2	139 (54.94)	643
(1, 1, 1/4)	261.25	$A_2B_3C_1D_1E_1$	(35,20,168)	100.7	97 (62.87)	1151	(22,38,160)	110.6	100 (61.72)	689
(1, 1, 1/8)	237.63	$A_2B_3C_1D_1E_2$	(27,19,227)	76.16	74.375 (68.70)	1182	(28,40,153)	90,48	87,125 (63.34)	724

system has not been implemented yet. So, it is assumed to be a single cell, which means EE is not applicable. The current state scheduling is done intuitively, with the number of voids and $C_{\max}$ values of 214 and 189, respectively. The performance of the proposed methods was evaluated on the average and best values obtained by repeatedly running the algorithms 3 times. The computational results are presented in Table 5. The value corresponding to the current company heuristic in the application ($Z_{Current}$) and the best values obtained with GA (Z_{GA}) and MPA (Z_{MPA}) were compared, and the deviation values (% improvement rates) were calculated as ($Z_{\text{Current}}-Z_{GA}$)*100/$Z_{Current}$ and as ($Z_{Current}-Z_{MPA}$)*100/$Z_{Current}$, respectively.

When the values in Table 5 are examined, the average (%) performance improvement values for both GA and MPA are quite close to each other compared to the current situation (57.40% and 56.73%, respectively). The greatest improvements were achieved in the fourth case (1,1,1/8 weight set), with values of 68.7% by the GA and 63.34% by MPA. Tables 6 and 7 show the results of scheduling and cell assignment related to the fourth case obtained by the GA and MPA.

As seen in Tables 6 and 7, three cells were created by both the GA and MPA. In both test problems and real-life applications, the proposed GA and MPA results show a very superior performance, especially in terms of CPU times, compared to the developed mathematical model.

6. Conclusion

In this paper, simultaneous consideration of Cell Formation (CF) and Cellular Part Scheduling (CPS) in the CMS environment is investigated. A new mathematical model has been developed by considering the assumptions and properties of the problem. The developed mathematical model offers a useful representation of the problem, and in terms of considered objective function components, it is the first study in the CF and CPS literature. The developed mathematical model has been tested on small-, medium-, and large-sized test problems derived from the literature.

The results have confirmed that the small sized problems can easily be solved with GAMS software. These optimum values are obtained in very short computational times. However, the time needed to reach the optimal solutions is rapidly growing as the size of the studied problem grows due to its NP-hard structure. On the other hand, most real-life CMS-CPS problems are generally larger in size than the instances handled here. So, to extend the applicability of the proposed model on much larger-sized and more complex problems, a Genetic Algorithm (GA) and an Marine Predators Algorithm (MPA) have also been developed and presented in the study. The GA parameters have been tuned using the Taguchi method. The results obtained on the generated medium- and large-sized problem sets show that both proposed GA and MPA have very effective performances, and reasonable and applicable solutions can be reached in acceptable shorter times. And finally, the proposed GA and MPA have been applied to a real-life case. The improvements obtained are fairly high, as expected.

Acknowldgement

The authors would like to express gratitude to Professor A. Attila Islier for his encouragement and valuable comments, which led to an improvement in this paper, and to Dr. Burak Urazel for his support regarding the coding of genetic algorithms. Research is supported by Eskisehir Osmangazi University-Turkey

Table 6. Genetic Algorithm (GA), real life application case 4, (a) Cell design and grouping and (b) Cell scheduling.

Cell no.	Part no.	Machine no.																							
		1	**6**	**7**	**13**	**14**	**21**	**2**	**3**	**4**	**5**	**9**	**10**	**12**	**15**	**16**	**18**	**19**	**22**	**23**	**24**	**8**	**11**	**17**	**20**
1	1	1	4	2		3	6								5				7						
	4	6			1	7	2											4			5				3
	11	1,7	4	2	5	3,8	6																		
	12	1	4	2		3	5																		
2	6											4	2			1		6	3	5					
3	2							3													2	4		1	
	3								5		2	1										3,6	4		
	5							3					4	1					5			7	6		2
	7											3										5	1		2,4
	8							4								1	6			5		8	2	7	3
	9												3	9					4,6			7	1,5	8	2
	10	4										2		3		6						7	1		5

(a)

	Sequence (Part. operation)		**Sequence (Part. operation)**
M_1	(1.1) →(12.1) →(11.1) →(10.4) →(4.6) →(11.7)	$M13$	(4.1)→(11.5)
$M2$	(5.3)→(8.4)→(2.3)	$M14$	(11.3)→(1.3)→(12.3) →(4.7)→(11.8)
$M3$	(3.5)	$M15$	(1.5)
$M4$	-	$M16$	(6.1)→(8.1) →(10.6)
$M5$	(3.2)	$M17$	(2.1)→(9.8) →(8.7)
$M6$	(1.4)→(12.4)→(11.4)	$M18$	(8.6)
$M7$	(1.2)→(11.2) →(12.2)	$M19$	(6.6)→(4.4)
$M8$	(3.3)→(5.7)→(9.7) →(3.6)→(2.4)→ (7.5)→(8.8)→(10.7)	$M20$	(4.3)→(5.2)→(7.2) →(9.2) →(7.4)→(8.3)→(10.5)
$M9$	(3.1)→(7.3)→(6.4) →(10.2)	$M21$	(4.2) →(1.6) →(11.6)→(12.5)
$M10$	(6.2)→(5.4)→(9.3)	$M22$	(5.5)→(6.3)→ (9.4)→(9.6)→(1.7)
$M11$	(7.1)→(9.1) →(10.1)→(9.5)→(8.2) →(5.6)→(3.4)	$M23$	(6.5)→(8.5)
$M12$	(5.1)→(10.3)→(9.9)	$M24$	(4.5)→(2.2)

(b)

Table 7. Marine Predators Algorithm (MPA) real-life application case 4 (a) Cell design and grouping and (b) Cell scheduling.

Cell no.	**Part no.**	**Machine no.**																							
		1	**4**	**8**	**10**	**12**	**22**	**6**	**7**	**9**	**13**	**17**	**18**	**19**	**20**	**21**	**23**	**2**	**3**	**5**	**11**	**14**	**15**	**16**	**24**
1	2	3		4								1					2								
	3	1	2	3	4														5			6			
	5				1,6	4	2,5			3												7			
	6	4				2	3							6		1									5
	7	3		5			2,4														1				
	9				5	1,3,9	4,6					8			2							7			
	10	2		7	1,3										5	6		4							
	11	1,7	2	3,8				4			5		6												
	12		2	3												5		1	4						
2	4									6	1	4			3	2	5					7			
	8				2					4			6	7	3		5					8		1	
3	1									1					7				4	2		3	5	6	

(a)

	Sequence (Part. operation)		**Sequence (Part. operation)**
$M1$	(11.1)→(10.2)→(3.1) →(7.3)→ (2.3)→(6.4)→(11.7)	$M13$	(4.1)→(11.5)
$M2$	(12.1)→(10.4)	$M14$	(1.3) →(9.7)→(4. 7)→(3.6)→(5.7)→(8.8)
$M3$	(1.4)→(12.4)→(3.5)	$M15$	(1.5)
$M4$	(11.2)→(12.2)→(3.2)	$M16$	(8.1)→(1.6)
$M5$	(1.2)	$M17$	(2.1)→(4.4)→(9.8)
$M6$	(11.4)	$M18$	(8.6)→(11.6)
$M7$	–	$M19$	(6.6)→(8.7)
$M8$	(12.3)→(11.3)→(3.3) →(2.4)→(11.8)→(7.5)→(10.7)	$M20$	(9.2)→(4.3)→(8.3) →(10.5)→(1.7)
$M9$	(1.1)→(8.4)→(5.3) →(4.6)	$M21$	(6.1) →(4.2)→(12.5)→(10.6)
$M10$	(10.1)→(8.2)→(10.3) →(5.1) →(9.5)→(3.4)→(5.6)	$M22$	(9.4)→(7.2) →(5.2)→(9.6)→(6.3) →(7.4)→(5.5)
$M11$	(7.1)	$M23$	(2.2)→(8.5)→(4.5)
$M12$	(9.1)→(9.3)→(6.2) →(5.4)→(9.9)	$M24$	(6.5)

(b)

(ESOGU) Research Projects Commission grant 201515A207 (2015-787).

References

1. Mahdavi, I., Aalaei, A., Paydar, M.M., et al. "Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems", *Int. J. Prod. Res.*, **49**(21), pp. 6517–6537 (2011). DOI: 10.1080/00207543.2010.524902

2. Buruk Sahin, Y. and Alpay, S. "A metaheuristic approach for a cubic cell formation problem", *Expert Syst. Appl.*, **65**, pp. 40–51 (2016). DOI: 10.1016/j.eswa.2016.08.034

3. Buruk Sahin, Y. and Alpay, S. "A new mathematical model for the integrated solution of cell formation and part scheduling problem", *Gazi Uni. J. Sci.*, **32**(4), pp. 1196–1210 (2019). DOI: 10.35378/gujs.471637

4. Pinedo, M.L., *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, USA (2002).

5. Li, D., Wang, Y., Xiao, G., et al. "Dynamic parts scheduling in multiple Jop shop cells considering intercell moves and flexible routes", *Comput Oper Res.*, **40**(5), pp. 1207–1223 (2013). https://doi.org/10.1016/j.cor.2012.11.024

6. Amjad, M.K., Butt, S.I., Kousar, R., et al. "Recent research trends in genetic algorithm based flexible jop shop scheduling problems", *Math. Probl. Eng.*, **2018**(8), p. 9270802 (2018). DOI: 10.1155/2018/9270802

7. Chaudhry, I.A. and Khan, A.A. "A research survey: review of flexible Jop shop scheduling techniques", *Int T Oper Res.*, **23**(3), pp. 551–591 (2016).

8. Wu, X., Chu, C., Wang, Y., et al. "A genetic algorithm for cellular manufacturing design and layout", *Eur. J. Oper. Res.*, **181**(1), pp. 156–167 (2007). https://doi.org/10.1016/j.ejor.2006.05.035

9. Wang, X., Tang, J., and Yung, K. "A scatter search approach with dispatching rules for a joint decision of cell formation and parts scheduling in batches", *Int. J. Prod. Res.*, **48**(12), pp. 3513–3534 (2010). DOI: 10.1080/00207540902922828

10. Ghezavati, V. and Saidi-Mehrabad, M. "Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time", *Int J Adv Manuf Tech.*, **48**(5), pp. 701–717 (2010).

11. Kesen, S.E., Das, S.K., and Güngör, Z. "A genetic algorithm based heuristic for scheduling of virtual manufacturing cells (VMCs)", *Comput Oper Res.*, **37**(6), pp. 1148–56 (2010). https://doi.org/10.1016/j.cor.2009.10.006

12. Tang, J., Wang, X., Kaku, I., et al. "Optimization of parts scheduling in multiple cells considering intercell move using scatter search approach", *J. Intell. Manuf.*, **21**(4), pp. 525–537 (2010).

13. Aryanezhad, M.B., Aliabadi, J., and Tavakkoli-Moghaddam, R. "A new approach for cell formation and scheduling with assembly operations and product structure", *Int. J. Ind. Eng. Comput.*, **2**(3), pp. 533–546 (2011). http://growingscience.com/ijiec/ijiec.html

14. Tang, J., Yan, C., Wang, X., et al. "Using lagrangian relaxation decomposition with heuristic to integrate the decisions of cell formation and parts scheduling considering intercell moves", *IEEE Trans. Autom. Sci. Eng.*, **11**(4), pp. 1110–1121 (2014). DOI: 10.1109/TASE.2014.2325860

15. Liu, C. and Wang, J. "Cell formation and task scheduling considering multi-functional resource and part movement using hybrid simulated annealing", *Int. J. Comput. Intell. Syst.*, **9**(4), pp. 765–777 (2016). DOI: 10.1080/18756891.2016.1204123

16. Rafiei, H., Rabbani, M., Gholizadeh, H., et al. "A novel hybrid SA/GA algorithm for solving an integrated cell formation job scheduling problem with sequence-dependent set-up times", *Int. J. Manag. Sci. Eng. Manag.*, **11**(3), pp. 134–142 (2016). DOI: 10.1080/17509653.2014.1003109

17. Liu, C., Wang, J., Leung, J.Y.T., et al. "Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm", *Int. J. Prod. Res.*, **54**(3), pp. 923–944 (2016). DOI: 10.1080/00207543.2015.1113328

18. Iqbal, N., Aziz, M.H., Jahanzaib, M., et al. "Integration of cell formation and job sequencing to minimize energy consumption with minimum make-span", *P I Mech Eng B-J Eng.*, **231**(14), pp. 2636–2651 (2017).

19. Costa, A., Cappadonna, F.A., and Fichera, S. "A hybrid genetic algorithm for minimizing makespan in a flow-shop sequence-dependent group scheduling problem", *J. Intell. Manuf.*, **28**(6), pp. 1269–1283 (2017). DOI: 10.1007/s10845-015-1049-1

20. Feng, H., Xia, T., Da, W., et al. "Concurrent design of cell formation and scheduling with consideration of duplicate machines and alternative process routings", *J. Intell. Manuf.*, **30**, pp. 275–289 (2019). DOI: 10.1007/s10845-016-1245-7

21. Far, M.H., Haleh, H., and Saghaei, A. "A fuzzy bi-objective flexible cell scheduling optimization model under green and energy-efficient strategy using Pareto-based algorithms: SATPSPGA, SANRGA, and NSGA-II", *Int J Adv Manuf Tech.*, **105**, pp. 3853–3879 (2019). DOI: 10.1007/s00170-019-03797-w

22. Forghani, K. and Ghomi, S.F. "Joint cell formation, cell scheduling, and group layout problem in virtual and classical cellular manufacturing systems", *Appl. Soft Comput.*, **97**(10), p. 106719 (2020). DOI: 10.1016/j.asoc.2020.106719

23. Halat, K. and Bashirzadeh, R. "Concurrent scheduling of manufacturing cells considering sequence-dependent family setup times and intercellular transportation times", *Int J Adv Manuf Tech.*, **77**(9-12), pp. 1907–1915 (2015). DOI: 10.1007/s00170-014-6511-2

24. Ebrahimi, A., Kia, R., and Komijan, A.R. "Solving a mathematical model integrating unequal-area facilities layout and part scheduling in a cellular manufacturing system by a genetic algorithm", *SpringerPlus*, **5**(1), p. 1254 (2016). DOI: 10.1186/s40064-016-2773-5

25. Subhaa, R., Jawahar, N., and Ponnambalam, S.G. "An improved design for cellular manufacturing system associating scheduling decisions", *Sādhanā*, **44**(7), p. 155 (2019). DOI: 10.1007/s12046-019-1135-8

26. Rahimi, V., Arkat, J., and Farughi, H. "A vibration damping optimization algorithm for the integrated problem of cell formation, cellular scheduling, and intercellular layout", *Comput Ind Eng.*, **143**, p. 106439 (2020). DOI: 10.1016/j.cie.2020.106439

27. Neufeld, J.S., Teucher, F.F., and Buscher, U. "Scheduling flowline manufacturing cells with intercellular moves: non-permutation schedules and material flows in the cell scheduling problem", *Int. J. Prod. Res.*, **58**(21), pp. 6568–6584 (2020). https://doi.org/10.1080/00207543.2019.1683251

28. Alimian, M., Ghezavati, V., and Tavakkoli-Moghaddam, R. "New integration of preventive maintenance and production planning with cell formation and group scheduling for dynamic cellular manufacturing systems", *J. Manuf. Syst.*, **56**, pp. 341–358 (2020). https://doi.org/10.1016/j.jmsy.2020.06.011

29. Wang, J., Liu, C., and Zhou, M. "Improved bacterial foraging algorithm for cell formation and product scheduling considering learning and forgetting factors in cellular manufacturing systems", *IEEE Syst J.*, **14**(2), pp. 3047–3056 (2020). DOI: 10.1109/JSYST.2019.2963222

30. Rafiee, M., Kayvanfar, V., Mohammadi, A., et al. "A robust optimization approach for a cellular manufacturing system considering skill-leveled operators and multi-functional machines", *Appl. Math. Model.*, **107**, pp. 379–397 (2022). DOI: 10.1016/j.apm.2022.02.028

31. Goli, A., Tirkolaee, E.B., and Aydın, N.S. "Fuzzy integrated cell formation and production scheduling considering automated guided vehicles and human factors", *IEEE Trans Fuzzy Syst*, **29**(12), pp. 3686–3695 (2021). DOI: 10.1109/TFUZZ.2021.3053838

32. Kazemi, M., Sadegheih, A., Lotfi, M.M., et al. "Developing a bi-objective schedule for an online cellular manufacturing system in an MTO environment", *Soft Comput.*, **26**(2), pp. 807–828 (2022). DOI: 10.1007/s00500-021-06402-z

33. Holland, J.H. "Adaptation in natural and artificial systems. an introductory analysis with applications to biology, control and artificial intelligence", Ann Arbor: University of Michigan Press (1975).

34. Shapiro, J. "Genetic algorithms in machine learning", In *Advanced Course on Artificial Intelligence*, pp. 146–168, Springer, Berlin, Heidelberg (1999).

35. Grefenstette, J.J. "Optimization of control parameters for genetic algorithms", *IEEE T Syst Man Cy.*, **16**(1), pp. 122–128 (1986). DOI: 10.1109/TSMC.1986.289288

36. Gen, M. and Cheng, R., *Genetic Algorithms and Manufacturing Systems Design*, John Wiley and Sons, Inc (1996).

37. Gopalakrishnan, H. and Kosanovic, D. "Operational planning of combined heat and power plants through genetic algorithms for mixed 0-1 nonlinear programming", *Comput Oper Res.*, **56**(3), pp. 51–67 (2015). DOI: 10.1016/j.cor.2014.11.001

38. Chakrabortty, R. and Hasin, M. "Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach", *Int J Ind Eng Comp.*, **4**(1), pp. 1–12 (2013). DOI: 10.5267/j.ijiec.2012.09.003

39. Faramarzi, A., Heidarinejad, M., Mirjalili, S., et al. "Marine predators algorithm: A nature-inspired metaheuristic", *Expert Syst. Appl.*, **152**, 113377 (2020). DOI: 10.1016/j.eswa.2020.113377

40. Hu, G., Zhu, X., Wei, G., et al. "An improved marine predators algorithm for shape optimization of developable Ball surfaces", *Eng. Appl. Artif. Intell.*, **105**(10), p. 104417 (2021). DOI: 10.1016/j.engappai.2021.104417

41. Gonggui, C., Ying, X., and Fangjia, L. "An improved marine predators algorithm for short-term hydrothermal scheduling", *IAENG Int. J. Appl. Math.*, **51**(4) (2021).

42. Fattahi, P., Saidi Mehrabad, M., and Jolai, F. "Mathematical modeling and heuristic approaches to flexible jop shop scheduling problems", *J. Intell. Manuf.*, **18**(3), pp. 331–42 (2007). DOI: 10.1007/s10845-007-0026-8

43. Candan, G. and Yazgan, H.R. "Genetic algorithm parameter optimisation using Taguchi method for a flexible manufacturing system scheduling problem", *Int J Prod Res.*, **53**(3), pp. 897–915 (2015). DOI: 10.1080/00207543.2014.939244

44. Far, M.H., Haleh, H., and Saghaei, A. "A flexible cell scheduling problem with automated guided vehicles and robots under energy-conscious policy", *Sci. Iran.*, **25**(1), pp. 339–358 (2018). DOI: 10.24200/SCI.2017.4399

45. Esmailnezhad, B. and Saidi-Mehrabad, M. "A two-stage stochastic supply chain scheduling problem with production in cellular manufacturing environment: A case study", *Sci Iran*, **30**(4), pp. 1399–1422 (2023). DOI: 10.24200/sci.2021.53506.3277

46. Cheng, L., Tang, Q., Zhang, L., et al. "Scheduling flexible manufacturing cell with no-idle flow-lines and job-shop via Q-learning-based genetic algorithm", *Comput. Ind. Eng.*, 108293 (2022). https://doi.org/10.1016/j.cie.2022.108293

47. Majumdar, A. and Ghosh, D. "Genetic algorithm parameter optimization using Taguchi robust design for multi-response optimization of experimental and historical data", *Int. J. Comput. Appl.*, **127**(5), pp. 26–32 (2015). DOI: 10.5120/ijca2015906383

Table A.1. Data for real-life problem.

Part no.	Operation no.	Operation type	Alternative Machines	Operation time (minutes)
1	1	Turning	1,2,9	35,46,38
	2	Hobbing	4,5,7	42,35,50
	3	Washing	8,14	7,9
	4	Trimming	3,6	8,10
	5	Carburising	13,15	15,18
	6	Grinding	16,18,21	34,36,42
	7	Marking	20,22	10,12
2	1	Hobbing	17,19	38,45
	2	Fitting the bush	23,24	10,15
	3	Turning the bush	1,2	6,9
	4	Washing	8,14	23,25
3	1	Turning	1,9	10,15
	2	Hobbing	4,5,7	34,36,43
	3	Washing	8,14	12,14
	4	Milling	10,11,12	12,16,18
	5	Trimming	3,6	4,6
	6	Washing	8,14	10,13
4	1	Carburising	13,15	15,17
	2	Grinding	16,18,21	5,8,9
	3	Marking	20,22	4,7
	4	Honing	17,19	11,16
	5	Fitting the bush	23,24	5,8
	6	Turning the bush	1,9	14,16
	7	Washing	8,14	10,11
5	1	Milling	10,11,12	6,8,11
	2	Drilling	20,22	3,5
	3	Turning	1,2,9	7,9,10
	4	Milling	10,12	14,17
	5	Drilling	20,22	3,4
	6	Hobbing	10,11	8,9
	7	Washing	8,14	10,12
6	1	Grinding	16,18,21	24,32,29
	2	Surface Milling	10,11,12	16,21,23
	3	Drilling	20,22	5,7
	4	Notching	1,9	3,2
	5	Fitting the bush	23,24	4,6
	6	Honing	17,19	11,14

48. Urazel, B. and Buruk Sahin, Y. "Solving a cubic cell formation problem with quality index using a hybrid meta-heuristic approach", *Gazi Univ. J. Sci.*, **36**(2), pp. 752–771 (2023). DOI: 10.35378/gujs.1003331

49. Ross, P.L., *Taguchi Techniques for Quality Engineering*, McGraw-Hill Book Company, New York (1988).

Appendix A.1

Table A.1 is represented in this appendix.

Table A.1. Data for Real-life Problem (continued).

Part no.	Operation no.	Operation type	Alternative Machines	Operation time (minutes)
7	1	Milling	10,11	60,52
	2	Drilling	20,22	10,12
	3	Turning	1,9	24,35
	4	Drilling	20,22	5,6
	5	Washing	8,14	11,12
8	1	Surface grinding	16,18,21	4,8,6
	2	Surface milling	10,11	25,18
	3	Chamfering	20,22	5,8
	4	Turning	2,9	17,22
	5	Fitting the bush	23,24	6,8
	6	Finish grinding	16,18	11,13
	7	Honing	17,19	10,14
	8	Washing	8,14	14,12
9	1	Rough milling	10,11,12	22,18,26
	2	Drilling	20,22	9,10
	3	Final milling	10,12	16,17
	4	Chamfering	20,22	5,3
	5	Hobbing	10,11	6,7
	6	Drilling	20,22	11,9
	7	Washing	8,14	12,14
	8	Honing	17,19	10,15
	9	Plug fitting	10,12	5,8
10	1	Grading	10,11	17,15
	2	Turning	1,2,9	27,35,36
	3	Milling	10,11,12	22,18,24
	4	Boring	1,2	24,30
	5	Deep drilling	20,22	10,15
	6	Grinding	16,18,21	8,12,15
	7	Washing	8,14	11,13
11	1	Turning	1,2,9	17,22,25
	2	Hobbing	4,5,7	10,15,13
	3	Washing	8,14	13,11
	4	Trimming	3,6	8,13
	5	Carburising	13,15	11,14
	6	Grinding	16,18,21	6,8,9
	7	Turning the bush	1,9	21,26
	8	Washing	8,14	5,9
12	1	Turning	1,2,9	6,7,11
	2	Hobbing	4,5,7	16,18,14
	3	Washing	8,14	4,7
	4	Trimming	3,6	5,9
	5	Grinding	16,18,21	26,24,28

Biographies

Yeliz Buruk Sahin received her PhD degree in Industrial Engineering from Eskisehir Osmangazi University, Turkey. She has been working as an Assistant Professor in the same department since 2017. She has several national and international publications in well-known high-impact journals in the fields of expert systems, environmental engineering, and material science. Her areas of research interest include Response Surface Methodology, Scheduling, Metaheuristic Algorithms, and Optimization.

Serafettin Alpay received his PhD from Eskisehir Osmangazi University, Department of Industrial Engineering, in 2003. He is currently working as an Associate Professor in the same department. His areas of interest are Production Management, Scheduling, Decision Support Systems, Meta-Heuristic Algorithms, Multi-Criteria Decision Making, and Artificial Intelligence Algorithms.
He has many publications scanned in different journals and in different field indexes like SCI, SCI-Exp., Scopus, etc.