Solving Joint Job Shop Scheduling of Production System Using Meta-heuristic Algorithm

Sandhya R Mishra[1], Madhurima Peram[2], Saurabh Pratap[3],Dnyaneshwar Mogale[2]
[1,2,3]*Indian Institute of Information Technology, Design & Manufacturing, Jablpur, MP India*
[4]*Indian Institute of Technology, Kharagpur, West Bengal*

***Abstract*— This paper addressed the job-shop scheduling problem and formulated a model as mixed integer programming to minimize the processing time of job. Each job has fixed machining operations, which has to perform simultaneously with reducing the total make-span. The machining process consist various complexities and uncertainties during the machine processing time, which is the challenging task to capture in the model. Due to NP hard nature of job shop scheduling problem, we have proposed two meta-heuristic algorithms; Cuckoo Search Algorithm (CSA) and Chemical Reaction Optimization to solve the mathematical model. We have tested the model in three different problem instances and found that CRO performs better than CSA, but there is a trade-off in convergence, the CSA convergence rate is better than CRO.**

***Keywords*— Job-shop scheduling, MIP, Cuckoo Search Algorithm, Chemical Reaction Optimization**

I. Introduction

The Job shop scheduling plays a major role in the growth of manufacturing firms. Due to uncertainties in the machining process, job scheduling to the available machine are the major challenging task, which enforce the researchers to focus and identify the crucial factors associated with job shop scheduling. The job has fixed machining operations and each has to execute on giving resource availability within defined duration of time. Job shop scheduling consists of the machines and jobs in which the jobs are scheduled in different machines with proper utilization and optimization of machine operations [1] [2]. In most of the manufacturing sectors, job shop scheduling plays a key role, i.e. in automotive industries, the different parts have different machining process, and the assembly stations assemble the final product after receiving all the parts. So, the assembly point requires all the completed jobs/parts in minimum make span. The machine has operation on which it is processed in a given time window [3]. The proper utilization of resources to optimize the job shop scheduling decrease energy consumption [4]. The main challenge is to determine the sequence of resources and minimize the job completion time. The manufacturing firms are focused to improve the productivity of machine through proper utilization of resources and minimizing the make span of the production cycle.

The paper is structured as follows. In section 2, we have described the past literature review on single objective and multi-objective job shop scheduling problem. Section 3 presents the problem statement and proposed mathematical model. The Solution approach has been described in section 4. In section 5, the result has been analyzed. In last section, we have concluded with the future scope of research.

II. Literature Review

Pempera & Smutnicki (2018) [5] have introduced the open cyclic policy and used permutation-graph-technology to determine minimum cycle criterion. Li et al. (2018) [6] described a problem, where the job arrives at flow shops at uncertain time period. The model is formulated to reduce the total completion time by using heuristic approach. Kundakcı & Kulak et al. (2016) [7] proposed a dynamic job shop scheduling problem and used the hybrid genetic algorithm to minimize the make-span. Yazdani et al (2017) [8] formulated a job shop scheduling problem as mixed integer linear programming (MILP) and maximize the sum of earliness-tardiness of jobs. The Hybrid imperialist competitive algorithm is proposed to determine the optimal schedule of jobs at the flow-shop. The artificial neural network [9] is introduced to determine the dynamic job shop scheduling and determine the output as optimal jobs schedule at machine shop.

Job shop scheduling consists of the machines and jobs in which the jobs are scheduled in different machines with proper utilization and optimization of machine operations. The machine has operation on which it is processed in a given time window [3]. Bürgy & Bülbül (2018) [10] dealt with non-regular function of job shop scheduling with operations start time and difference between the capricious pairs of jobs operations. The proposed model minimizes the defined the convex cost functions and minimizes the make-span. The Rule based heuristic has been proposed to determine the optimal scheduling of ships with optimal allocations of berth [11].

Some of the researchers focused the multi-objective job shop scheduling problems and described the tradeoff objectives. The job shop scheduling with human error and preventive maintenance has been described [12] and formulated the multi-objective model to capture the tradeoff between make span of jobs, human error and machine availability for job processing. Torkashvand et al. (2017) dealt with a flow shop scheduling problem and formulate the model as multi-objective, which minimize the make-span and total tardiness [13].

The most of researchers focused on job shop scheduling problems and formulated as mixed integer linear programming MILP. In this paper, we have formulated the model as mixed integer programming and used two nature inspired; Cuckoo Search and Chemical Reaction Optimization Algorithms.

III. Problem Statement

A Job Shop is a work location for performing a variety of operations using a number of general purpose machines. Each job has to undergo a sequence of operations and, the machines for performing each operation are selected accordingly. In this paper, we consider P jobs and Q operations to be performed on different jobs. R machines are available in the workstation which can be used to process each job. Different operations performed on the work are denoted by T_{prq} , which represents operation q performed on the job p using machine r. The duration for each of these operations is represented by E_{prq}. Other than the operation time, each machine is also associated with a fixed setup time depending on the work and the operation to be performed, represented by U_{prq}.

Different machines have different work cycles. Hence, each machine is provided with a time period beyond which it has to be taken into maintenance. During this period, the jobs will not be processed in that machine and the jobs have to wait for a fixed amount of time depending on the respective maintenance periods of the machines. There are some constraints that have to be followed while performing the operation in a Job Shop which is discussed in the section. Here, our objective is to obtain the best sequence for the operations through which all the jobs can be processed using the available machines in the least amount of time. Fig 1 represents the flow of work taking place in a Job Shop.

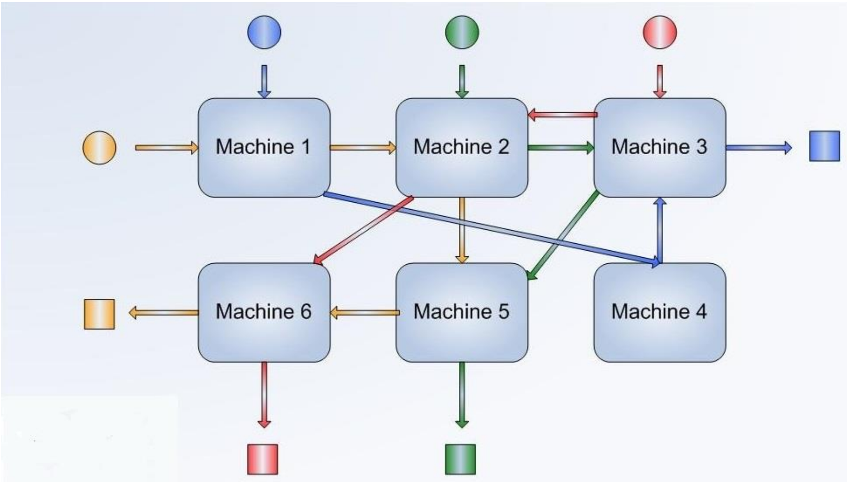

Figure 1: Schematic diagram showing the flow of work in a Job Shop

Sets

P	Set of Jobs
R	Set of Machines
Q	Set of operations

Indices

p	= {1,2,…..P}	Jobs
r	= {1,2,….R}	Machines
q	= {1,2,…..Q}	Operations

Parameters

W	Fixed maintenance time for each machine
X_r	Average life of each machine
E_{prq}	Processing time for each operation
U_{prq}	Setup time associated with each operation

Variables

STO_{prq}	Start time of each operation
DL_r	Delay produced on each machine

The Objective function:
Minimize:

$$\left[\sum_{prq}\left(E_{prq}+U_{prq}+DL_r\right)\right] \quad \textbf{(1)}$$

Constraints:

$$STO_{prq} \geq STO_{p(r-1)q}+E_{p(r-1)q} \qquad \forall p \in P, r \in R, q \in Q \quad \textbf{(2)}$$

$$[STO_{prq}, STO_{prq}+E_{prq}] \cap [STO_{p'r'q'}, STO_{p'r'q'}+E_{p'r'q'}] = \phi \qquad \forall\ p,p' \in P,\ r,r' \in R,\ q,q' \in Q \quad \textbf{(3)}$$

$$DL_r = W \qquad \forall\ STO_{prq} \geq X_r \quad \textbf{(4)}$$

In the objective function (1), term 1 gives the time involved in processing the jobs, term 2 and term 3 represents start time and the setup time of the machine respectively and the term 4 gives the delay produced due to each machine entering its maintenance phase. Constraint (2) is called the sequence constraint. It ensures that the production schedule remains feasible. The resource constraint for the problem under consideration is represented by the constraint (3). It ensures that the same machine is not working on two different jobs at the same time. Constraint (4) gives the maintenance constraint. It gives the duration for which a machine would work before it enters the maintenance phase. Here, we try to optimize the time involved in such a way that least time is lost during the whole process.

IV. Solution Methodology

The major objective of this work is to minimize the maximum duration(C), which is the progressive time to complete all operations on all machines with the given sequence, resources and the constraints respectively. For this JSSP, a mathematical relation of Process time (E_{prq}), Starting time of each operation (STO_{prq}), set up time (U_{prq}), and delay produced in each operation (DL_r)) is formulated considering P jobs and R machines and Q operations to minimize the maximum duration. In this paper, two meta-heuristic approaches are proposed to get the instantaneous best optimal solution. These two techniques are:

1. Cuckoo Search Optimization Algorithm
2. Chemical Reaction Optimization Algorithm

4.1 Cuckoo Search Optimization algorithm

The Cuckoo Search Algorithm is an optimization algorithm developed by Yang and Deb based on the obligate brood parasitic behavior of cuckoos [14].

CSA is inspired by the breeding behavior of cuckoos. Some species of cuckoos, for instance, the ani and Guira, engage special reproduction strategy. They lay eggs in the nest of other birds. The responsibility of nurturing the young is thus offloaded to the host birds. The host birds, somehow, are able to detect the cuckoo eggs at times. This prompts the cuckoos to evolve eggs with higher similarity to their hosts. In order to explain the CSA, there are three idealized rules that should be considered:

- It is assumed that each cuckoo lays one egg at a time and dumps it randomly in a host nest.
- The nests with high quality eggs will be passed on to the next generation.
- The number of available host nests is fixed, while the probability of a cuckoo egg can be discovered as an alien egg is *Pα*. The host bird can either throw the alien egg or abandon the nest and build a completely new nest in a new location.

The objective is always to use the better solution (cuckoos) and replace the previous not-so-good solution. One of the main reasons for the popularity of CS is its global convergence [14]. Studies proved that CS has characteristics of the global optimizer with ability for converging to the true global optimum and hence gives optimum search values.
The standard deviations *u* and *v* are represented as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\lambda) \quad (5)$$

$$Levy \approx u = t^{-\lambda}, \ (1 < \lambda < 3) \quad (6)$$

$$s = u \times |v|^{-1/\beta} \quad (7)$$

$$u \approx N(0.\sigma_v^2) \quad (8)$$

$$v = N(0.\sigma_v^2) \quad (9)$$

$$\sigma_v = \left\{ \frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(0.5 + \frac{\beta}{2}\right) \beta 2^{\frac{\beta-1}{2}}} \right\}^{1/\beta} \quad (10)$$

$$\sigma_v = 1 \quad (11)$$

In Equation (5) x_i^t represents the present solution and x_i^{t+1} represents the new solution for cuckoo *i*, when the L´evy flight is performed. λ is the parameter which helps the cuckoo in finding more locations and σ is the probability density function. The CSA can be summarized by pseudo code, given in Figure 2

1. *Initialized: I,P,K,T*
2. *Objective function* ***f(x), x=(x₁,x₂,…………..x_d)ᵀ****,*
3. *Generate initial population of n host nests* ***x_i , (i=1,2,…….,n)****.*
4. *While****(t <MaxGen)*** *or (termination criteria: TC)*
5. *Get the j cuckoo randomly by Levy flight.*
6. *Evaluate its quality by the given objective function* ***F_i****.*
7. *Choose a nest* ***j*** *among* ***n*** *nests randomly*
8. *If* ***(F_i < F_j)***
9. *Replace* ***j*** *by new solution*
10. *end*
11. *A fraction* ***(P_a)****of worse nests are abandoned,*
12. *And build new ones at new locations via Levy flight.*
13. *Keep best solution (or nests with quality Solutions).*
14. *Rank the solutions and find the current best*
15. *end*
16. *Report the best solution found*

Figure 2: Pseudo-code for CS algorithm

4.2 Chemical Reaction Optimization Algorithm

The population based meta-heuristic algorithm named chemical reaction optimization (CRO) first developed by Lam and Li. CRO is the method which pair the phenomena of chemical reaction with optimization to get the best solution. The conversion process of a molecule from highly energetic unstable to a stable state during a chemical reaction is embedded inside the code of CRO, to solve optimization problem [15].

CRO involves pool of molecules to explore the solution space and each molecule contains a set of attribute given in Table 1.These attributes are formulated to avoid the problem of getting stock in finding local optimal solution and also these affect the convergence speed of CRO. Four reaction operators involve in CRO are on-wall ineffective collision, decomposition, inter-molecular ineffective collision and synthesis[15]. Table 1 describes the algorithm parameters, which has mathematical means in algorithm.

Table 1: Algorithm Parameters of CRO

Symbol	*Chemical meaning*	*Mathematical meaning*
W	Molecular structure	Solution
PE	Potential energy	Objective function value
KE	Kinetic energy	Measure of tolerance of having worse solutions
NumHit	Number of hits	Current total number of moves
MinPE	Minimum value	Current optimal function value
MinHit	Minimum hit number	Move number at current optimal solution

4.2.1 On-wall ineffective collision

This effect occurs when the highly energetic molecule hits the wall of the container and then bounces back. Structure of the molecule will change after this collision. If the present structure of the molecule is ***w*** then after collision it will be ***w'***. The equation for the change in the structure of the molecule is given in Equation 12.

$$w(i) \rightarrow w'(i) \qquad (12)$$

Where, $w'(i) \in$ Neighborhood ($w(i)$).

After collision some part of the kinetic energy of the molecule will be lost and that energy will be stored in the chemical energy of the buffer. Applying the principle of conversion of energy the energy equation for the molecule is given in Equation 13 and 14.

$$K_{ew} = (P_{ew} - P_{ew'} + K_{ew'}) \times \alpha \qquad (13)$$

$$\sum_{i=1}^{k} PE_{w_i} + \sum_{i=1}^{k} KE_{w_i} - \sum_{i=1}^{t} PE_{w'_i} \geq 0 \qquad (14)$$

Where, KE_w = Kinetic energy of ***w***
PE_w = Potential energy of ***w***
$PE_{w'}$ = Potential energy of ***w'***
$KE_{w'}$ = Kinetic energy of ***w'***
α = Random number lies between [KElossrate, 1]

4.2.2 Decomposition

Decomposition operator of CRO refers to the problem when the molecule hits the wall and break down into two or more molecules. This collision is so vigorous that the final molecular structure will be different from the original one. Due to this decomposition some part of the energy of ***w*** gets lost during the process when we finish the local search from ***w*** to ***w'*.** Due to conversion of energy ***w*** may sometimes does not have enough energy to sustain its transformation into ***w'(i)*.**So, a certain portion of the energy buffer accumulated during the on wall collision can be utilized to support the change. The *DecThres* is a threshold value of carrying on Decomposition operator.

4.2.3 Inter-molecular ineffective collision

This collision occurs when two molecules collide with each other and then get separated. Number of molecules after the collisions remain unchanged and it gives flexibility to the structure of the molecules. But, more the number of molecules is involved in the collision more will be the required energy. Hence, in the original code of CRO only two molecules are considered to be a part of inter-molecular ineffective collision.

4.2.4 Synthesis

Synthesis is the phenomena when two or more molecules collide and then combine to form one new single molecule. This process implies that the search regions are expanded. The *SynThres* is a threshold value of carrying on synthesis operator.
Figure 2 describes the pseudo code of the Chemical Reaction Optimization Algorithm

```
Start
Step-1: Define the optimization problem and initialize the parameters
Step-2: Initialize population of the molecule
Step-3: while the stopping criteria not meet do
        Generate b ϵ [0, 1]
        if  b > MolCol then
           Randomly select one molecule M
         goto Step-4
         Else goto Step-5
         End if
End while
Step-4 Uni-molecular operator
If  w(i).Numhit > w(i).Minhit > DecThres
     Decomposition operator of CRO ();
Else
      On-wall ineffective Operator of CRO ();
End if
       Update energy management rules();
Step-5: Inter-molecular operator
If (w(i).KE < SynThres) && (w'(i) < SynThres)
    Synthesis operator of CRO ();
Else
     Inter-molecular ineffective operator of CRO ();
End if
Step-6: Termination criterion.
Stop if the maximum generation number is achieved
       Otherwise, repeat from Step-3
End
```

Figure 2: Pseudo-code for CRO Algorithm

V. RESULT AND DISCUSSION

The proposed model tries to reduce the manufacturing lead time associated with the Job Shop taken into consideration. The solution approach aims at determining the least possible lead time for the production operation addressed in this paper. The mathematical model developed in this paper is solved using two different algorithms-CRO and Cuckoo Search. The code for both the algorithms was generated in MATLAB R2014a in a Windows 10 environment, respectively, on a computer with Intel Core i5, 2.9GHz processor, and 8GB RAM. Different parameters used and the values assigned to them are shown in this section. The solution has been obtained after executing both the algorithm for 10 times.
Table 2 gives the values associated with CRO meta-heuristics and Table 3 gives values of CSA met-heuristics used in this paper. Different parameters used in the mathematical formulation are given in Table 4. The CRO and CSA are run for a particular set of data and the performance of the algorithms is analyzed based on the results obtained.

Table 2 Algorithm Parameters (CRO)

Parameters	CRO
Popsize	4
KElossrate	0.2
InitialKE	40.36×10^6
molColl	0.2
SynThres	15
DecThres	800
Alpha	3
Beta	40.34×10^6

Table 3 Algorithm Parameters (CSA)

Parameters	CSA
Popsize	25
σ_v	1
P_a	0.25
λ	1 to 3

Table 4 Parameters

Parameter	Value
W	200
P	3
R	3
Q	2
X_1, X_2, X_3	80, 80, 80

Here, the Job Shop is considered to have 3 jobs, 3 machines and 2 operations associated with it. Table 5 gives the processing time involved with each operation T_{prq} taking place in the Job shop. The setup time involved with each operation is given in Table 6.

Table 5 (Values of E_{prq})

(:,:,1)		r_1	r_2	r_3	(:,:,2)		r_1	r_2	r_3
	p_1	58	30	49		p_1	27	28	55
	p_2	59	41	42		p_2	43	27	45
	p_3	33	35	47		p_3	34	57	50

Table 6 (Values of U_{prq})

(:,:,1)		r_1	r_2	r_3	(:,:,2)		r_1	r_2	r_3
	p_1	58	56	44		p_1	44	60	59
	p_2	52	51	44		p_2	44	50	41
	p_3	51	43	54		p_3	53	53	57

Table 7 Computation Experiment

Case	Case Type (P-R-Q)	Algorithm	Solution	Generations	Computational Time
1	(3-3-2)	CRO	485	166	5.324 s
		Cuckoo	506	120	15.735 s
2	(5-5-3)	CRO	731	139	7.431 s
		Cuckoo	1306	77	19.975 s
3	(7-7-4)	CRO	1682	107	10.278 s
		Cuckoo	3202	100	24.679 s

Table 7 reveals the computation experiment for three different case scenario of the considered model, and Fig 5, Fig 6, Fig 7 gives the corresponding convergence graphs of CRO for case-1, case-2 and case-3 respectively, and Fig 8, Fig 9, and Figure 10 gives the corresponding convergence graphs of CSA for case-1, case-2 and case-3 respectively.

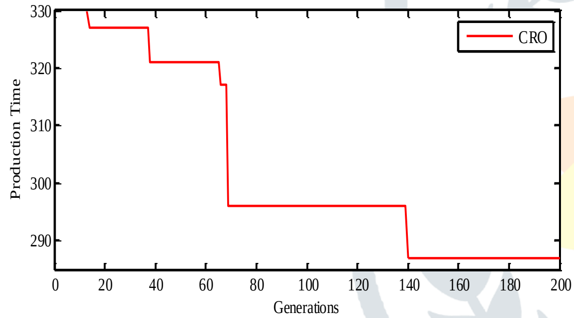

Figure 5: Convergence graph for CRO of case-1

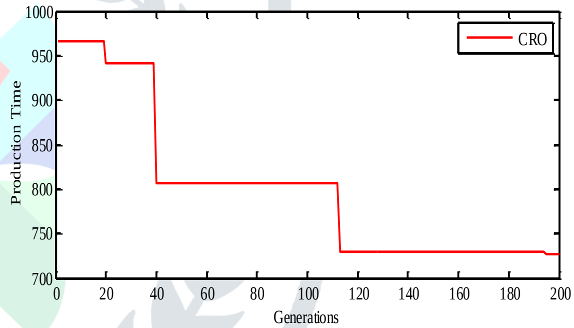

Figure 6: Convergence graph for CRO of case-2

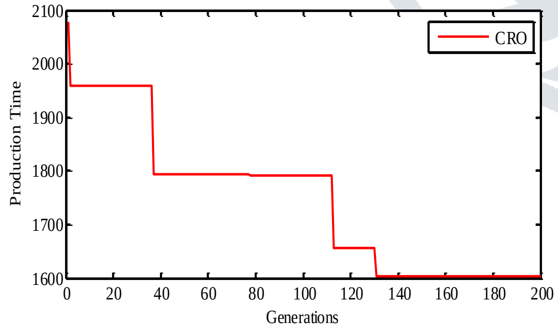

Figure 7: Convergence graph for CRO of case-3

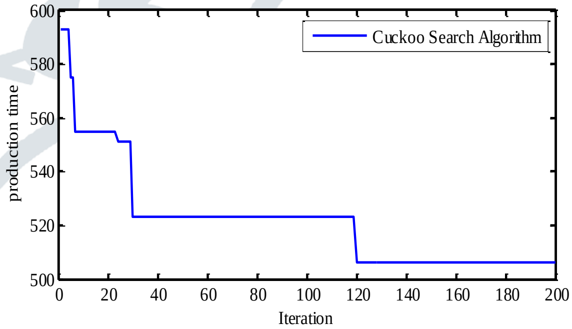

Figure 8: Convergence graph for CSA of case-1

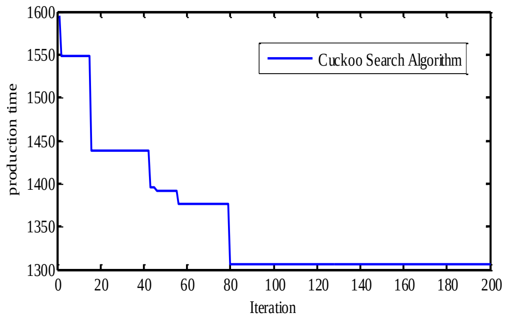

Figure 9: Convergence graph for CSA of case-2

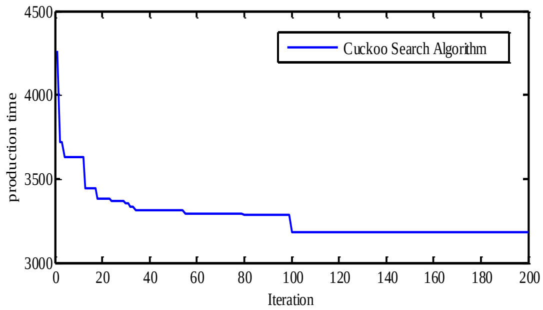

Figure 10: Convergence graph for CSA of case-3

VI. CONCLUSION

In this paper, a mathematical model is formulated to minimize the time taken to complete all operations on all machines with the given sequence, and associated resources and the constraints. This model is solved using two meta-heuristic algorithms namely Chemical Reaction Optimization (CRO) and Cuckoo Search Algorithm (CSA), both the algorithms were found to be effective in solving the objective of this paper. The model has tested in three different problem instances; found that CSA gives better solution for the model while CRO gives better convergence time.

The possible avenue for future research is that the proposed model can be extended with parameters like delay caused during job transfer between workstation and assembly point .This can also be formulated based on the bottleneck operation.

REFERENCES

[1] R. H. Huang, "Multi-objective job-shop scheduling with lot-splitting production," *Int. J. Prod. Econ.*, vol. 124, no. 1, pp. 206–213, 2010.

[2] C. Özgüven, L. Özbakir, and Y. Yavuz, "Mathematical models for job-shop scheduling problems with routing and process plan flexibility," *Appl. Math. Model.*, vol. 34, no. 6, pp. 1539–1548, 2010.

[3] J. W. Pawel SITEK, "APPLICATION OF DIFFERENT COMPUTATIONAL ENVIRONMENTS TO OPTIMIZATION OF MAKESPAN IN THE FLOW PRODUCTION LINE," *Appl. Comput. Sci.*, vol. 1, no. 1, pp. 219–238, 2005.

[4] L. Meng, C. Zhang, X. Shao, and Y. Ren, "MILP models for energy-aware flexible job shop scheduling problem," *J. Clean. Prod.*, vol. 210, pp. 710–723, 2018.

[5] J. Pempera and C. Smutnicki, "Open shop cyclic scheduling," *Eur. J. Oper. Res.*, vol. 269, no. 2, pp. 773–781, 2018.

[6] G. Li, N. Li, N. Sambandam, S. P. Sethi, and F. Zhang, "Flow shop scheduling with jobs arriving at different times," *Int. J. Prod. Econ.*, vol. 206, no. July, pp. 250–260, 2018.

[7] N. Kundakci and O. Kulak, "Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem," *Comput. Ind. Eng.*, vol. 96, pp. 31–51, 2016.

[8] M. Yazdani, A. Aleti, S. M. Khalili, and F. Jolai, "Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem," *Comput. Ind. Eng.*, vol. 107, pp. 12–24, 2017.

[9] S. K. Sim, K. T. Yeo, and W. H. Lee, "An expert neural network system for dynamic job shop scheduling," *Int. J. Prod. Res.*, vol. 32, no. 8, pp. 1759–1773, 1994.

[10] R. Bürgy and K. Bülbül, "The job shop scheduling problem with convex costs," *Eur. J. Oper. Res.*, vol. 268, no. 1, pp. 82–100, 2018.

[11] S. A. K. I. Babu *et al.*, "Minimizing delay of ships in bulk terminals by simultaneous ship scheduling, stockyard planning and train scheduling," *Marit. Econ. Logist.*, vol. 17, no. 4, pp. 464–492, 2015.

[12] M. Sheikhalishahi, N. Eskandari, A. Mashayekhi, and A. Azadeh, "Multi-objective open shop scheduling by considering human error and preventive maintenance," *Appl. Math. Model.*, vol. 67, pp. 573–587, 2019.

[13] M. Torkashvand, B. Naderi, and S. A. Hosseini, "Modelling and scheduling multi-objective flow shop problems with interfering jobs," *Appl. Soft Comput. J.*, vol. 54, pp. 221–228, 2017.

[14] X.-S. Yang and S. Deb, "Cuckoo Search via Levey Flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, 2009, pp. 210–214.

[15] S. Pratap, A. Nayak, A. Kumar, N. Cheikhrouhou, and M. K. Tiwari, "An integrated decision support system for berth and ship unloader allocation in bulk material handling port," *Comput. Ind. Eng.*, vol. 106, pp. 386–399, 2017.