International Journal of Industrial Engineering & Production Research March 2023 Vol. 34, No. 2: 1-20
DOI: 10.22068/ijiepr.34.2.11

RESEARCH PAPER

A Mathematical Model for Double Resource Constraint Flexible Job-shop Scheduling Problem Considering the Limit of Preventive Maintenance

Amir Nayeb[1], Esmaeil Mehdizadeh[2*] & Seyed Habib A. Rahmati[3]

Received 8 August 2022; Revised 20 December 2022; Accepted 23 January 2023;

ABSTRACT
In the field of scheduling and sequence of operations, one of the common assumptions is the availability of machines and workers on the planning horizon. In the real world, a machine may be temporarily unavailable for a variety of reasons, including maintenance activities, and the full capacity of human resources cannot be used due to their limited number and/or different skill levels. Therefore, this paper examines the Dual Resource Constrained Flexible Job Shop Scheduling Problem (DRCFJSP) considering the limit of preventive maintenance (PM). Due to various variables and constraints, the goal is to minimize the maximum completion time. In this regard, Mixed Integer Linear Programming (MILP) model is presented for the mentioned problem. To evaluate and validate the presented mathematical model, several small and medium-sized problems are randomly generated and solved using CPLEX solver in GAMS software. Because solving this problem on a large scale is complex and time-consuming, two metaheuristic algorithms called Genetic Algorithm (GA) and Vibration Damping Optimization Algorithm (VDO) are used. The computational results show that GAMS software can solve small problems in an acceptable time and achieve an accurate answer, and also meta-heuristic algorithms can reach appropriate answers. The efficiency of the two proposed algorithms is also compared in terms of computational time and the value obtained for the objective function.

KEYWORDS: *Flexible job-shop scheduling; Dual resource constraint; Preventive maintenance; Genetic algorithm; Vibration damping optimization algorithm.*

1. Introduction

Today's competitive market has taken steps to meet the diverse needs of customers, and this change has led to the creation of diverse and flexible equipment, but this equipment alone is not responsive to the competitive environment and is in dire need of planning and scheduling; So scheduling plays a vital role in the manufacturing industry [1, 2]. Scheduling issues in real manufacturing environments are often faced with unavailability of resources and complexity [3]; One of the scheduling issues is the flexible job shop problem. (FJSP) [4]. In recent decades, the issue of flexible job shops has been considered by most researchers due to its many applications in management and industry [5]. For example, it is used in the production of semiconductors, some petrochemical industries, glass industries, etc. In FJSP, only machine capacity is considered as a constraint factor of resources and workers' capacity is ignored as an effective factor in increasing productivity and production efficiency [6]. Therefore, the study of flexible job shop with dual constraint resources limited to humans and machines seems extremely necessary. Figure 1 schematically shows the Dual Resource Constrained Flexible Job Shop Scheduling Problem (DRCFJSP) considering the

* **Corresponding author:** ***Esmaeil Mehdizadeh***
es.mehdizadeh@iau.ac.ir

1. *Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*
2. *Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*
3. *Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*

limit of preventive maintenance. In comparison to FJSP, the DRCFJSP contains a new sub-problem called the worker selection problem[7]. Regarding the existing literature, the job shop scheduling problem is known as an NP-hard problem [8]. Therefore, it can be concluded that DRCFJSP can be provable as an NP-hard problem. In the DRCFJSP literature, several main and somewhat unrealistic assumptions are considered, which make the main issue far from the real world. One of these hypotheses is continuous and unlimited access to machines. By increasing the use of machinery, the reliability of the devices can be reduced and broken down. Unexpected breakdowns increase production costs [9], As a result, the use of maintenance methods prevents unexpected breakdowns and reduces production costs[10]. Maintenance activities play an important role in companies' decisions about costs, device reliability, or even product quality [11]. Scheduling in maintenance means deciding which maintenance activity to carry out and when to do so that the predetermined goals are optimal [12]. Scheduling in maintenance is a vital and inevitable issue. One of the known strategies today is preventive maintenance, which includes all actions that are planned and implemented to prevent the shot down of machines. The purpose of preventive maintenance is time-based maintenance (TBM). In time-based maintenance, maintenance activities are performed at the same periods, called periodic time [13]. This paper examines the solving of the Dual Resource Constrained Flexible Job Shop Scheduling Problem (DRCFJSP) considering the constraint of preventive maintenance in order to obtain the best sequence of actions, machine allocation and workers' selection.

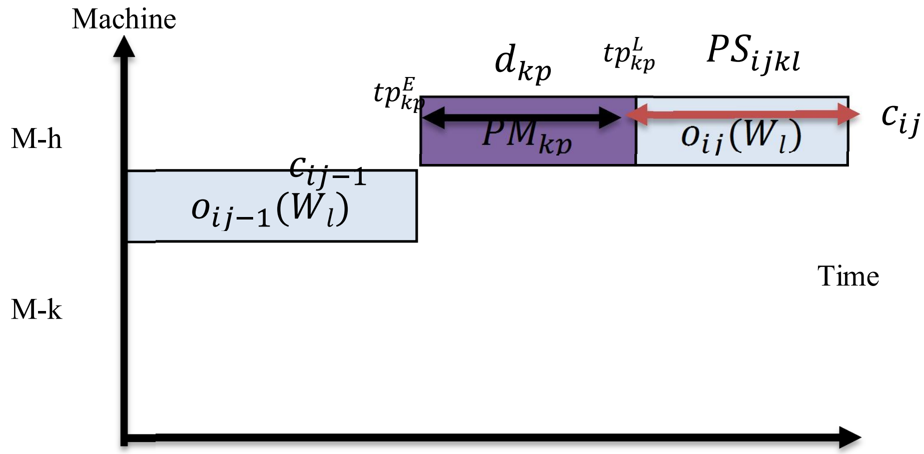


Fig. 1. Schematically shows the dual resource constrained flexible job shop scheduling problem (DRCFJSP) considering the limit of preventive maintenance.

2. Literature Review

Existing assumptions, such as the availability of machines at all times, have led to scheduling models being far from reality. This paper focuses on this gap and tries to fill part of it.

[14] examines job shop flow scheduling considering the machine access constraints. He assumed that, in the definitive model, periods of unavailability are predetermined; More precisely, an operation on the machine begins only if the processing is finished before the unavailability of the machine. In this research, two different approaches to preventive maintenance are presented. In the first approach, the place of unavailable periods is fixed and the aim is to find an optimal or near-optimal schedule with non-segregated operations. The second approach is that; the constraints of non-segregated operations may create idle time before any preventive action. Therefore, to minimize these idle times, it is assumed that maintenance periods are not fixed, but should be performed in definite time windows during scheduling. Aggoune proves that programs that have the same sequence on all machines may not necessarily contain the optimal answer. Finally, he used two genetic (GA) and Taboo Search (TS) algorithms to solve this problem[14-15]. studied the issue of flexible job shop scheduling with availability constraints. Availability constraint in this model is proposed to be non-fixed; in other words, the time to complete the maintenance actions is not fixed and should be determined during scheduling. Then they used the hybrid genetic algorithm to solve

flexible job shop scheduling problems with non-fixed availability constraints[15-16]. examined the issue of job shop scheduling in a sequence dependent on set-up time and preventive maintenance policies. They set the optimization criterion for the completion time of jobs and solved the problem using meta-innovative methods. [16-17]. investigated the minimization of total completion time in job shop scheduling by considering machine unavailability during the time planning horizon. In his article, he used graphs to illustrate the concept of blocks in periods of machine unavailability [17-18]. used a Pareto-based hybrid genetic algorithm to solve the problem of dual resource flexible job shop. They assumed the modal as mono-objective and bi-objective. Their goal was the shortest time to complete the actions and the cost of production[18-19]. proposed a new multi-objective mathematical model for the flexible job shop scheduling problem with parallel machines and maintenance costs. After modeling, two meta-heuristic algorithms are used, one is a hybrid genetic algorithm and the other is simulation annealing (SA). Finally, they compared the answers using two algorithms[19-20]. studied a single-machine scheduling problem by considering the processing time of jobs and multiple preventive maintenance. They considered three objective functions: Completion time, total completion time, and total weighted completion time. Because the problem was NP-hard, they used innovative algorithms and genetic algorithms to solve the problem[20-21]. proposed a variable neighborhood search algorithm (VNS) for a dual resource flexible job shop constraint for humans and machines to minimize completion time [21-22], presented simulated annealing (SA) algorithm and vibration damping optimization (VDO) algorithm for a dual resource flexible job shop constraint to humans and machines to obtain the minimum makespan [22-25]. implemented the same things using migrating birds' algorithm, knowledge guided fruit fly optimization algorithm (KGFOA) and Genetic algorithm came to acceptable conclusions[23-26]. solved the problem of condition-based maintenance (CBM) using flexible job shop action (FJSP). This problem has considered two modes of maintenance, one is corrective maintenance (CM) and the other is preventive maintenance (PM). They compared the value of corrective and preventive maintenance. In addition, to make the issue more realistic, they made it possible for the system to break down between inspection periods, and also the time and duration of maintenance have been randomly determined. To solve this model, they used a harmony search optimization algorithm[26], [27]. proposed a discrete particle swarm algorithm based on the maximum fit function for the problem, which aimed to minimize production time and cost [27], [28]. examined the issue of production scheduling on a single-machine model with preventive maintenance during the period in which the release date of jobs was considered. It should be noted that both resumable and non-resumable items were studied. The first part is provable using the earliest release date (ERD) which is polynomial time, and the second part can prove to be NP-hard; therefore, according to the mixed integer programming model (MIP) and using the earliest release date with the longest processing time (ERD-LPT) and branch and bound algorithm, they solved this problem[28], [29]. presented the DRCFJSP by considering green production indices and proposed a new hybrid genetic algorithm (NHGA) to solve this problem[7], [29]. using the memetic algorithm (MA) investigated solving multi-objective flexible job-shop scheduling problems with worker flexibility (MO-FJSPW). Their purpose in designing this issue is to minimize the maximum completion time, maximize the workload of the machines and obtain the sum of the workload of the machines[29].

[30-31] examined the same problem with multiple purposes[30, 32]. investigated the dual resources job shop scheduling problem using the indices of minimum completion time and due date and solved the problem using the greedy heuristics algorithm[32-33]. studied dual flexible job shop scheduling constrained to worker and machine considering the shutdown of machines, and because they considered the processing time uncertain, they presented a robust fuzzy-stochastic programming model (RFSP). Their purpose was to minimize the completion time of jobs[33-34]. examined DRCFJSP using a particle swarm optimization algorithm to minimize the time and cost[5]. [34]. proposed a model to solve DRCFJSP using artificial intelligence (AI)-based DRCFJSP optimization model. This model introduces the differences between the loading and unloading operation times of workers before and after the process. They used the quantum genetic algorithm (QGA) to minimize the maximum completion time [34-35]. proposed a model for solving the dual resource-constraint flexible job-shop scheduling problem with sequencing flexibility (DR-FJSPS). Their goal is

to minimize the makespan, maximal worker workload, and weighted tardiness. To solve this problem they use a non-dominated sorting genetic algorithm (NSGA-II) [35-36]. studied the flexible job shop scheduling problem with parallel machines by considering cleaner production criteria, dual human-machine resources, job release date, and machine speed-dependent processing time. The objective functions of this problem include minimizing the sum of earliness and tardiness and the speed increasing. They solve the model of a mixed integer programming using a Non-dominated Ranked Genetic Algorithm (NRGA) and compared the results with the non-dominated sorting genetic algorithm (NSGA-II) [36].

According to mentioned literature, there is a significant gap in most of the examined issues. One of these new opportunities in the scheduling literature is the dual-resource-constrained flexible job shop (DRCFJSP), and considering the existing literature and our best knowledge, the development of DRCFJSP is not covered by preventive maintenance. Therefore, in this study, DRCFJSP is investigated using maintenance. In the following and section 3, the problem definition, hypotheses, and mathematical model will be presented. Section 4 describes the proposed method to solve the problem, using a genetic algorithm and vibration damping optimization algorithm. In Section 5, to evaluate the performance of the proposed algorithms, problems in small, medium, and large dimensions are designed and solved by the mentioned algorithms and their results are presented. Section 6 provides conclusions and suggestions for future research.

3. Mathematical Model

In this problem, a mixed integer linear programming model was used to schedule the DRCFJSP. In this model, there are *n* independent jobs $J = \{j_1, j_2, \dots, j_n\}$, *l* human powers $W = \{w_1, w_2, \dots, w_l\}$ and *m* machines $M = \{M_1, M_2, \dots, M_m\}$. Each job like j_i contains *r* sequences of operations $\{o_{i1}, o_{i2}, \dots, o_{ir}\}$ ‘, which according to the precedence constraint, each of the operations o_{ij} (operation j of work i) is processed one after the other on a machine from a set of eligible machines (M_{ij}). Each machine is also assigned to a worker who is selected among a set of eligible workers (W_{ij}).

3.1. Hypotheses

1. All machines and workers are available from zero time.
2. A machine can only do one operation or one maintenance activity at a time.
3. The worker can be transferred from one machine to another but cannot leave the machine during the processing of an operation.
4. Interruption is not allowed.
5. An operation can be performed by different machines and workers, while the processing time will be different and definite.
6. Each process needs two resources machine and a worker.
7. Each worker can work on more than one machine, and each machine can be controlled by different workers.
8. Maintenance times are defined by a predetermined time window.
9. Machine setup time ignored.
10. Machines cannot do anything while maintaining that machine.

3.2. Indices

Jobs index $i, r = 1,2, \dots, n$
Machines index $k, h = 1,2, \dots, m$
Operation index $j, s = 1,2, \dots, n_i$
Workers index $l, b = 1,2, \dots, u$
Maintenance activity index $p = 1,2, \dots, p_j$

3.3. Parameters and sets

n: total number of jobs
m: total number of machines
n_i: total number of job operations
u: total number of workers
p_j: total number of maintenance activity
O_{ij}: operation *j* on the job *i*
PS_{ijkl} : processing time O_{ij} on machine *k* by worker *l*
AC_{ij} : A set of machines that are capable of processing O_{ij}
AE_{ij} : A set of skillful workers to do the operation O_{ij}
CE_{ijl}: A set of machines on which the O_{ij} the operation can be performed by worker *l*.
CB_{ijk} : A set of workers capable of doing O_{ij} operation on machine *k*
PM_{kp} : the *p* activity of maintenance on *k* machine
d_{kp} : the duration of maintenance activity PM_{kp}

$[tp_{kp}^{E}, tp_{kp}^{L}]$: the window time allocated to PM_{kp}, so that tp_{kp}^{E} is the earliest setup time and tp_{kp}^{l} is the latest end time of maintenance

M: a very large number

3.4. Decision variables

$X_{ijrs} = \begin{cases} 1 & \text{If } O_{ij} \text{ operation is done after } Ors \text{ operation} \\ 0 & \text{otherwise} \end{cases}$

$Y_{ijkl} = \begin{cases} 1 & \text{If the } Oij \text{ operation is done on machine } k \text{ by worker } l \\ 0 & \text{otherwise} \end{cases}$

$XP_{ijkp} = \begin{cases} 1 & if\ Oij \text{ operation is done after } PM_{kp} \\ 0 & \text{otherwise} \end{cases}$

C_{ij} : completion time of O_{ij}

Z_{kp} : completion time of maintenance activity PM_{kp}

C_{MAX} : maximum completion time of jobs

3.5. Mathematical model

$$Min\ C_{MAX} = \max_{1\le i\le n}\{C_{in_i}\} \tag{1}$$

Subject to:

$$\sum_k^m \sum_l^u Y_{ijkl} = 1, \quad where\ \forall i,j; i = 1,2,\dots,n; j = 1,2,\dots,m; \tag{2}$$

$$C_{ij} \ge \sum_k^m \sum_l^u PS_{ijkl} Y_{ijkl} \quad where\ \forall i,j; i = 1,2,\dots,n ; j = 1 ; \tag{3}$$

$$C_{ij} \ge C_{ij-1} + \sum_k^m \sum_l^u PS_{ijkl} Y_{ijkl} \quad where\ \forall i,j; i = 1,2,\dots,n ; j \ge 2 ; \tag{4}$$

$$C_{ij} \ge C_{rs} + \sum_{l\epsilon CB_{ijk}} PS_{ijkl} Y_{ijkl} - A(1 - X_{ijrs}) - A\left(3 - \sum_{l\epsilon CB_{ijk}} Y_{ijkl} - \sum_{l\epsilon CB_{rskl}} Y_{rskl}\right)\ where\ \forall\ i < n ,j; r > i , s ; k\epsilon\{AC_{ij} \cap AC_{rs}\} ; \tag{5}$$

$$C_{rs} \ge C_{ij} + \sum_{l\epsilon CB_{rsk}} PS_{rskl} Y_{rskl} - A(X_{ijrs}) - A\left(3 - \sum_{l\epsilon CB_{ijk}} Y_{ijkl} - \sum_{l\epsilon CB_{rsk}} Y_{rskl}\right)\ where\ \forall\ i < n ,j; r > i , s ; k\epsilon\{AC_{ij} \cap AC_{rs}\} ; \tag{6}$$

$$C_{ij} \ge C_{rs} + \sum_{k\epsilon CE_{ijl}} PS_{ijkl} Y_{ijkl} - A(1 - X_{ijrs}) - A\left(3 - \sum_{k\epsilon CE_{ijl}} Y_{ijkl} - \sum_{k\epsilon CE_{rsl}} Y_{rskl}\right)\ where\ \forall\ i < n ,j; r > i , s ; l\epsilon\{AE_{ij} \cap AE_{rs}\} ; \tag{7}$$

$$C_{rs} \ge C_{ij} + \sum_{k\epsilon CE_{rsl}} PS_{rskl} Y_{rskl} - A(X_{ijrs}) - A\left(3 - \sum_{k\epsilon CE_{ijl}} Y_{ijkl} - \sum_{k\epsilon CE_{rsl}} Y_{rskl} - \sum_k^m \sum_h^m Q_{ikh}\right) where\ \forall\ i < n ,j; r > i , s ; l\epsilon\{AE_{ij} \cap AE_{rs}\} ; \tag{8}$$

$$Z_{kp} \ge tp_{kp}^{E} + d_{kp} \quad where\ \forall\ k ,p; k = 1,2,\dots,m ;\ p = 1,2,\dots,p_j ; \tag{9}$$

$$Z_{kp} \le tp_{kp}^{l} \quad where\ \forall\ k ,p; k = 1,2,\dots,m ;\ p = 1,2,\dots,p_j ; \tag{10}$$

$$Z_{kp} \ge C_{ij} + d_{kp} - A(XP_{ijkc}) - A\left(1 - \sum_{l\epsilon CB_{ijk}} Y_{ijkl}\right)\ where\ \forall\ i ,j ,k ,p ; i = 1,2,\dots,n; j = 1,2,\dots,m ; k = 1,2,\dots,m ;\ p = 1,2,\dots,p_j \tag{11}$$

$$C_{ij} \ge Z_{kp} + \sum_{l\epsilon CB_{ijk}} PS_{ijkl} Y_{ijkl} - A(1 - XP_{ijkc}) - A\left(1 - \sum_{l\epsilon CB_{ijk}} Y_{ijkl}\right)\ where\ \forall\ i ,j ,k ,p ; i = 1,2,\dots,n; j = 1,2,\dots,m ; k = 1,2,\dots,m ;\ p = 1,2,\dots,p_j; \tag{12}$$

$$C_{ij} \ge 0 \quad where\ \forall i,j; i = 1,2,\dots,n; j = 1,2,\dots,m; \tag{13}$$

$$Z_{kp} \ge 0 \quad where\ \forall\ k ,p; k = 1,2,\dots,m ;\ p = 1,2,\dots,p_j ; \tag{14}$$

$$tp_{kp}^{E}, tp_{kp}^{l} \ge 0 \quad where\ \forall\ k ,p; k = 1,2,\dots,m ;\ p = 1,2,\dots,p_j ; \tag{15}$$

$$X_{ijrs} , Y_{ijkl} , XP_{ijkp}\ \epsilon\ \{0,1\} \quad where\ \forall i,j,r,s,k,h,l,p \tag{16}$$

In this model, equation (1) is an objective function of the problem; Which minimizes the maximum makespan (C_{MAX}). Equation (2) specifies that, O_{ij} operation is processed by which machine and worker. Equation (3) ensures that, the makespan of the first operation (C_{i1}) is at least as long as the processing time of that job. Equation (4) requires the model to observe the precedence constraints, in other words, at most one operation of each job should be processed at a time. Equations (5) and (6) identify the time relationship of the operation of two different jobs if two operations are processed by the same machine. In this case, one machine can process at most, one operation at a time. Equations (7) and (8) are disjunctive constraints and specify that two operations are processed by one worker. In this case, a worker can process at most, one operation at a time. Equations (9) and (10) ensure that the makespan of maintenance activities is within its time window. Equations (11) and (12) prevent the overlap of operations and maintenance activities on the machine in question, in other words, at any time it is done on a machine or a maintenance activity or an operation. Relationships (13), (14), (15), and (16) also indicate the type of decision variable.

4. Suggested Solutions

Dual resource flexible job shop scheduling problem constraint to human and machine is known as an NP-hard problem. So, enlarging the dimensions of the problem cause to decrease in the efficiency of accurate solutions. Hence, most heuristic and meta-heuristic methods are more effective for solving medium and large-scale problems that usually occur in the real world [37]. Therefore, in this study, two genetic algorithms and vibration damping optimization are used as approximate algorithms to solve optimization problems.

4.1. Genetic algorithm (GA)

Many evolutionary algorithms have been used in recent years. A genetic algorithm is one of the evolutionary algorithms which is in the category of guided random search technique [38]. Researchers have been inspired by the random mutation of genes that occur in reproduction and have developed it into the solution space. In their achievement, each child is from a combination of two parents, and the intersection operator is considered one of the basic components of this search technique[39]. In the following, the design of GA for solving the problem is presented.

4.1.1. The way of displaying the answer or the structure of chromosome

Chromosomes are encoded answers and solve space points. In this paper, each chromosome is randomly generated to find a random initial population. To do this, for each operation a house and in each house, a quadruple string (i, j, k, l) is provided to represent the operation, in which *i* represents the job number, j represents the operation number, k represents the number of machines and l represents the number of workers. String length is equal to the total number of operations of jobs [40]. In an example of displaying the answer, a problem with four jobs, eight operations, three machines, and two workers is randomly considered as the first parent.

)3,1,2,1(	)2,1,1,2(	)1,1,2,1(	)3,2,3,2(	)3,3,3,2(	)1,2,2,1(	)1,3,2,1(	)2,2,1,1(	) 2,3,1,2(

Fig. 2. Displaying the answer

4.1.2. Initial population

Generational scattering prevents rapid convergence and local optimization. Therefore, we randomly generate the initial population (initial answers) to maintain the scatter of the answers in the solving space as much as possible.

4.1.3. Objective function

To calculate the objective function of each chromosome (solving space points), these encoded answers must be decoded. For this purpose, to overcome the limitation of not having access to machines due to maintenance, two vectors are defined and their values are random numbers between the earliest and latest time of completion of maintenance operations $(tp_{kp}^{E}, tp_{kp}^{L})$ and the difference between the corresponding elements of these two vectors is equal to the makespan of that operation (d_{kp}) and their completion time (C_{ij}) is also calculated. Then, having the completion time of all operations, the longest completion time of jobs is calculated as $Min\ C_{MAX} = \max_{1\leq i\leq n}\{C_{in_i}\}$

4.1.4. Fitness function (evaluation function)

Regarding the nature of the genetic algorithm, all evaluation functions should be maximized. Whenever our problem is minimization, we have to convert it to maximization to be consistent with the nature of the genetic algorithm. This is done by subtracting the value of the objective function from the big positive number M and the value obtained is called the fitness value. So, the fitness function is:

$$fit(i) = M - of(i) \tag{17}$$

4.1.5. Selection strategy

Selection is the process by which individuals in a generation are selected pairwise for mating. There are different approaches for selection in literature [41]. In this article, the roulette wheel is used, according to which people with higher fitness have a better chance of mating, and this is the basis of Darwin's theory. Applying the roulette wheel approach, two parameters are required: the probability of selection $P(i)$ and the cumulative probability $CP(i)$, which are obtained in equations (18) and (19).

$$P(i) = \frac{fit(i)}{\sum_{i=1}^{popsize} fit(i)} \tag{18}$$

$$CP(i) = \sum P(i) \tag{19}$$

Based on the roulette cycle approach, a random number from the uniform distribution *R~U[0,1]* is generated, and then to select the answer the following rule is used:

$$R \leq CP(1) \rightarrow Chrom(1), \ldots, CP(i-1) < R < CP(i) \rightarrow Chrom(i) \tag{20}$$

4.1.6. Crossover

After selecting a pair of parents with one of the selection methods, the genetic operator of the crossover with a probability P_c is used to combine the two parents and produce two children [42]. The main problem of the crossover operator is that; the feasibility of newly generated answers may not be guaranteed. In cases where the answers by new chromosomes are not feasible, corrective action is usually taken to turn them into feasible answers, which will prolong the solving time[6]. In this paper, we use two crossover operators with the preservability of new chromosomes of each generation and no need for a modification process; one is improved precedence operation crossover (IPOX) for changing the sequence of generated answers, and the other is a multipoint preservative crossover for changing the worker and allocating machine[30]. The IPOX crossover operator is explained in three steps:

First step: a set of jobs is randomly selected and saved.

Second step: a set of selected jobs is copied from parent 1 to child 1, and from parent 2 to child 2.

Third step: The set of jobs not copied in step 2 is copied from parent 1 to child 2 and from parent 2 to child 1, except that the worker index and the allocated machine to child 1 are inherited from parent 1, and vice versa.

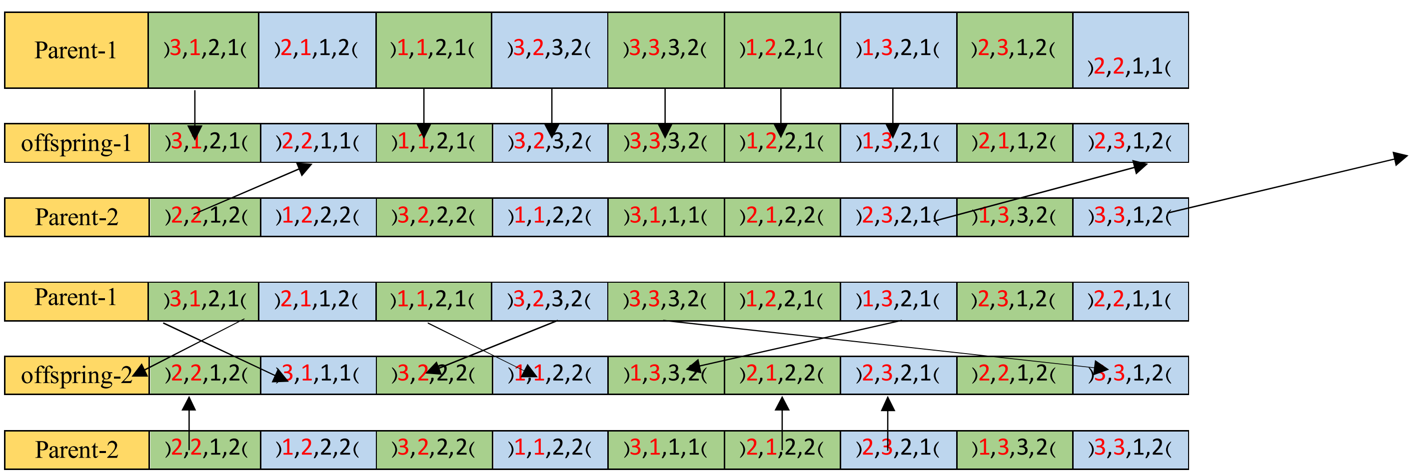


Fig. 3. IPOX

Multipoint preservative crossover is the same as IPOX, except that in each repetition in the operator, a uniform random vector is generated for a parent pair.

Parent-1	)3,1,2,1(	)2,1,1,2(	)1,1,2,1(	)3,2,3,2(	)3,3,3,2(	)1,2,2,1(	)1,3,2,1(	)2,3,1,2(	)2,2,1,1(
offspring-1	)3,1,2,1(	)2,1,2,2(	)1,1,2,1(	)3,2,3,2(	)3,3,1,2(	)1,2,2,1(	)1,3,2,1(	)2,3,2,1(	)2,2,1,2(
Random	0	1	0	0	1	0	0	1	1

Vector									
offspring-2	)2,2,1,2(	)1,2,2,1(	)3,2,2,2(	)1,1,2,2(	)3,1,2,1(	)2,1,2,2(	)2,3,2,1(	)1,3,2,1(	)3,3,3,2(
Parent-2	)2,2,1,2(	)1,2,2,2(	)3,2,2,2(	)1,1,2,2(	)3,1,1,1(	)2,1,2,2(	)2,3,2,1(	)1,3,3,2(	)3,3,1,2(

Fig. 4. MPX

4.1.7. Mutation

To prevent the production of unjustified answers and lead the children to the local optimal answer, the mutation operator with probability P_m is applied to each of the generated children. To do this, four-sequence neighborhood structures are used SNS-1، SNS-2 ، ANS-1 , ANS-2[43]:

The **SNS-1** neighborhood structure focuses on changing the sequence of operations in the generated solutions in such a way that, the allocation of operations to machines and workers does not change. In short, changes can be made by switching two adjacent operations, belonging to two different jobs that are randomly selected.

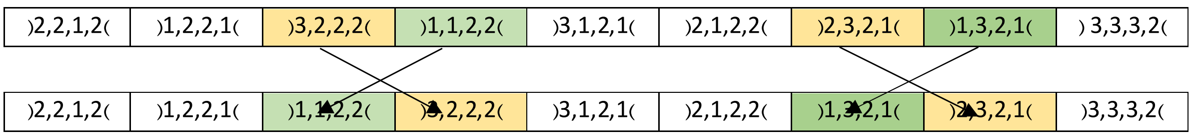


Fig. 5. SNS-1

The **SNS-2** neighborhood structure is used to replace operations related to two jobs in the generated answer. Implementing this structure, the allocation of operations to machines and workers doesn't change. In short, it can be said that two jobs are selected randomly, and changing the position of all operations of these two jobs, brings bigger changes.

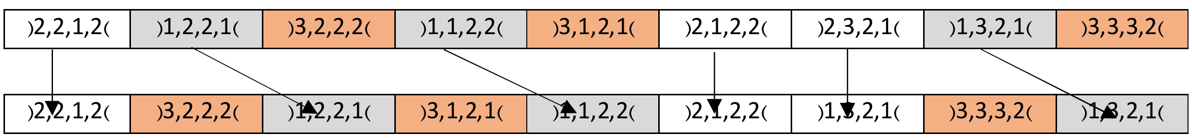


Fig. 6. SNS-2

ANS-1 neighborhood structure is used to change the allocation of operations to machines in generated answers, in such a way that, allocated workers and the sequence of operations on the machines do not change. In summary, the selected operation is randomly allocated to another machine among the eligible machines for that operation.

O_{22} O_{33}

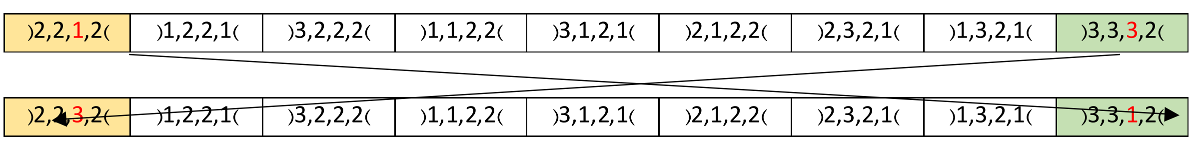


Fig. 7. ANS-1

ANS-2 neighborhood structure is used to change the allocation of operations to workers in generated answers, in such a way that, allocated machines and the sequence of operations on the machines do not change. Briefly, an operation is randomly selected and arbitrarily allocated to another worker among the eligible workers of that operation.

O_{22} O_{33}

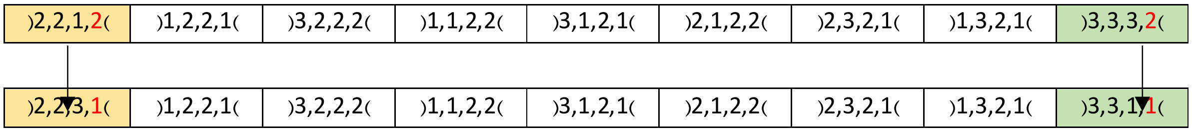


Fig. 8. ANS-2

So, the first and second mutations (SNS-1 and SNS-2) were used to optimize the sequence of operation, and the third and fourth mutations (ANS-1 and ANS-2) were used to allocate operations to machines and workers.

4.1.8. Stopping criteria

In this algorithm, going through a certain number of repetitions of offspring is considered a stopping criterion. Finally, the pseudo-code of the genetic algorithm is as follows.

[Initialization]
[Initialize Parameters] (PopSize, Numgen, Pc, Pm, StopCriteria,..)
[Initialize Population] Generate PopSize chromosomes, randomly.
[Evaluation] Evaluate the fitness of each chromosome.
[New Generation]
Repeat
[Selection] Select Parents based on a selection strategy.
[Crossover]Produce (PopSize * Pc) of offspring with Crossover.
[Mutation] Produce (PopSize * Pm) of offspring with Mutation.
[Reproduction] Copy remaining chromosomes based on elitism.
[Replacing] Place new offspring in the new population.
[Evaluation] Evaluate the fitness of each chromosome.
Until StopCriteria is met
[End] Return the best solution to the final population

4.2. Vibration damping optimization algorithm

One of the meta-heuristic algorithms for solving NP-hard problems is the vibration damping optimization algorithm. The algorithm is inspired by physical systems in which vibration is defined as an oscillating or periodic movement around the equilibrium point of an object, where the position of the object is obtained when no force exerts on it [3]. All objects that have mass and elastic properties can have vibrational movement. Every vibrational movement is caused by a force within the system or a force. In oscillating systems, part of the energy of the system is always wasted in the form of heat and sound. The amplitude gradually decreases over time and eventually, the oscillator stops oscillating. This process is called damping[44]. The more the amplitude, the more the frequency of answers. In other words, in high amplitude, due to the larger domain, a larger range or space is at hand and newer answers are more likely to occur. Conversely, at low amplitude, a new response is less likely to occur; And when the amplitude tends to zero, the system stops oscillating [45]. This algorithm was presented for the first time by Mehdizadeh and Tavakkoli Moghaddam for single-objective problems[46]. The performance of the algorithm is in such a way that; The move starts with a random initial answer, generates a random answer in each repetition, and uses possible rules to search the neighborhood[47].

In this section, the VDO algorithm is used for solving DRCFJSP. Figure 9 shows the process of the suggested VDO algorithm. The VDO algorithm starts with generating random answers. Then the algorithm parameters include the initial domain (A_0), minimum domain (A_{min}), number of searches (repetitions) per domain (L), damping coefficient (γ), number of damping cycles (t), and Rayleigh distribution parameter or standard deviation (σ) are quantified. Then the answers are evaluated by the values of the objective function (E) [3]. It should be noted that the method of displaying the answer is quite similar to the genetic algorithm.

$t = 0$, $A = A_0$, $X_{best} = \phi$, Generate X_0 , $X_{best} = X_0$
DO (Damping Loop)
$n = 0$
Do (Force Loop)
Select a move randomly and run over X_n as: $\Delta E = E(X_{new}) - E(X_{last})$
If $\Delta E < 0$ then $X_{best} = X_{new}$ and n=n+1 and $X_n = X_{new}$
Else
Generate $r \rightarrow U[0,1]$ Randomly, Set $z = 1 - e^{-A^2/2\sigma^2}$
If $r < z$ then $n = n + 1$ & $X_n = X_{new}$, End if
Loop while $n < N$
$t = t + 1$, $A_t = A_0 e^{\frac{-\varphi t}{2}}$
Loop While ($t < T$ and $A_t > 0$)
Print X_{best}

Fig. 9. Pseudo-code vibration damping optimization algorithm

In the first step, the desired parameters (γ and σ ،L ،A_0) are obtained and the number of repetitions in each domain (L) and the number of damping cycles are equal to zero. In the second step, a random initial answer is generated (X_0) and the value of its objective function $E(X_0)$ is calculated. In the third step, it is achieved with a new answer (X_{new}) using the neighborhood structure and the value of its objective function $E(X_{new})$ is also calculated. And the best of them is selected (X_{best}), i.e., the new answer replaces the previous one; If the following relationship is:

$$\Delta E = E(X_{new}) - E(X_{last}) < 0 \quad (21)$$

If ΔE> 0, the number r is randomly generated between (1 and 0), and if the following relation is established, the new answer replaces the previous one.

$$r < 1 - \exp(-\frac{A^2}{2\sigma^2}) \quad (22)$$

Otherwise, the new answer is rejected and a unit is added to the number of repetitions per domain (L). If ($L < L_{max}$) is not established, it returns to step three, but if ($L < L_{max}$) is established, a unit is added to the number of damping cycles (t) and the stop criterion is checked. If the following relation is not established, it returns to step three and the algorithm continues.

$$A_t = A_0 \exp(-\frac{t}{q}) \quad (23)$$

In the presented VDO algorithm, the considered stopping criterion is to implement the non-improvement algorithm in the objective function.

5. Mathematical Results

Consider the environment of DRCFJSP concerning preventive maintenance. In this section, a problem including 3 jobs, 3 machines, and 2 workers is examined by considering the maintenance time window to minimize the maximum makespan. Tables 2 and 3 show the processing time of operation and implementing time of maintenance activities respectively.

Tab. 2. processing time of operation

	Machine	M1		M2		M3	
	Worker	W1	W2	W1	W2	W1	W2
OPERATIONS	O11	12	20	8	20	10	20
	O21	6	7	9	10	20	7
	O31	20	20	4	9	20	8
	O21	16	13	10	11	10	15
	O22	11	17	20	20	20	14
	O23	7	8	5	8	6	8
	O31	20	7	9	8	20	20
	O32	20	10	20	11	20	12
	O33	20	18	20	16	20	15

Tab. 3. implementing time of maintenance activities

Machine	PM_{kp}	tp_{kp}^{E}		tp_{kp}^{L}		d_{kp}	
		E1	E2	E1	E2	E1	E2
1	PM_{1p}	7	12	12	18	4	6
2	PM_{2p}	6	9	9	12	3	3
3	PM_{3p}	5	10	11	15	5	5

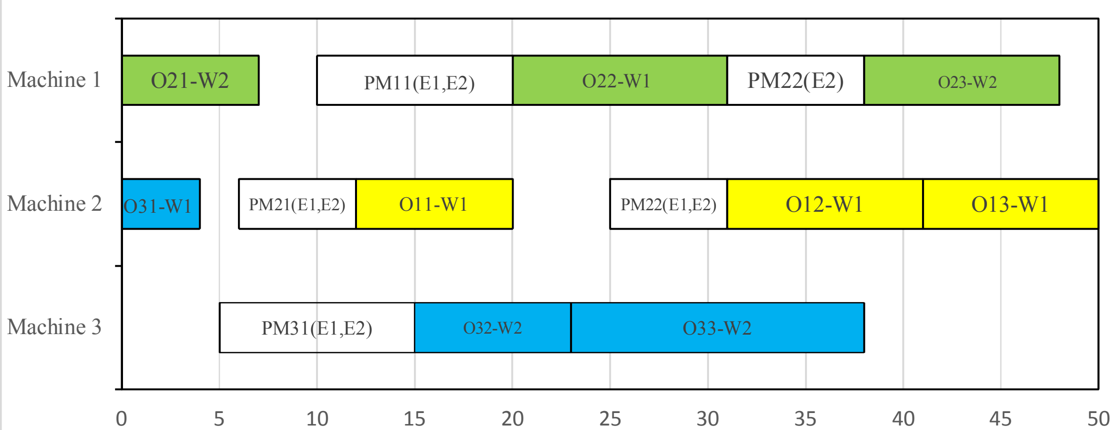


Fig. 10.Gant chart

As far as the mathematical model of this problem is mixed integer linear programming, it is used the CPLEX Solver which is in the GAMS software. By solving the above problem, the start and end times of each job are obtained according to the purpose of the problem and the above problem achieves the optimal global answer in less than 200 seconds. In this section, the Gantt diagram related to the problem is drawn according to the results. As shown in Figure 10, the maximum termination time (C_{MAX}) is 50.

To increase the efficiency of the algorithm and consider the sensitivity of the algorithms to the input parameters, in this research the input parameters are set to their best value using the Taguchi method. To evaluate the performance of the presented algorithms, 27 problems in small, medium, and large dimensions have been designed and solved by the suggested algorithms; also, these problems have been resolved by GAMS software as much as possible. 3600 seconds (equivalent to 1 hour) is considered a stop condition for GAMS software in solving medium and large problems. In other words, the answers obtained from GAMS software after 3600 seconds are compared with the answers obtained from two genetic and vibration damping optimization algorithms. Table 5 shows the generated problems and the results of comparing the genetic algorithm, vibration damping optimization, and CPLEX solver in GAMS software. As shown in this table, in some problems the genetic algorithm and in others the vibration damping optimization algorithm provided better answers and did not follow a specific pattern; But in small problems, GAMS software provides accurate answers. Regarding the response duration, in small problems, the speed of the genetic algorithm is higher than the vibration damping algorithm, and by contrast, the speed of the vibration damping optimization algorithm is higher in large problems. All calculations of genetic and vibration damping optimization algorithms are programmed in MATLAB 2018a and are performed on a personal computer with a frequency of 2.6 GHz (Core i5) and RAM of 6 GB. Also, the optimal parameter levels obtained using Minitab19 are shown in Table 4.

Tab. 4. Optimal parameter levels

Solving Methodology	Parameter	Optimum Value (Small)	Optimum Value (Medium)	Optimum Value (Large)
GA	Npop	40	60	100
	Pc	0.85	0.9	0.8
	Pm	0.2	0.15	0.25

VDO	A_0	10	15	20
	L	1	1.5	2
	σ	150	200	250
	γ	0.1	0.15	0.2

Tab. 5. comparison table of GA and VDO answers with CPLEX solver.

Test Problem					GAMS (CPLEX Solver)		GA		VDO	
NO	n	m	u	p	C_{MAX} Average (in 5 runs)	Run Time(sec)	C_{MAX} Average (in 5 runs)	Run Time(sec)	C_{MAX} Average (in 5 runs)	Run Time(sec)
SMALL Problem Size										
S1	3	3	2	2	50	198	50	52	50	90
S2	4	4	3	2	67	201	67	53	67	90
S3	5	4	4	2	81	216	81	61	81	97
S4	6	5	3	2	97	240	97	79	97	110
S5	6	5	4	2	113	263	113	87	113	112
S6	7	5	3	3	169	301	169	124	169	151
S7	8	5	4	3	202	345	202	137	202	169
S8	9	6	4	3	294	410	294	191	294	209
S9	9	7	6	3	486	893	486	265	486	379
MEDIUM Problem Size										
M1	11	9	7	4	721	986	762	416	745	441
M2	13	10	10	4	937	1342	1039	627	1106	525
M3	15	13	12	4	1246	1566	1351	833	1444	998
M4	17	15	15	4	2039	2164	2231	1001	2335	1054
M5	18	16	15	5	2781	3306	3081	1674	2984	1431
M6	19	16	15	5	3004	3600	3515	1832	3366	1710
M7	19	18	17	5	3420	3600	3929	2149	4023	1982
M8	20	18	17	5	4197	3600	4197	2743	4301	2509
M9	20	19	18	5	4949	3600	4791	3006	4580	2781
LARGE Problem Size										
L1	30	20	20	6	---	3600	6941	4219	7210	3811
L2	35	25	25	7	---	3600	8199	4371	8393	4099
L3	40	30	30	8	---	3600	9214	4794	9140	4317
L4	45	35	35	9	---	3600	10541	5016	10499	4588
L5	50	40	40	10	---	3600	12737	5642	13121	5212
L6	55	45	45	11	---	3600	15789	6001	16124	5791
L7	60	50	50	12	---	3600	18354	6435	20019	6099
L8	65	55	55	13	---	3600	21743	7091	22142	6749
L9	70	60	60	14	---	3600	26456	8869	27187	8039

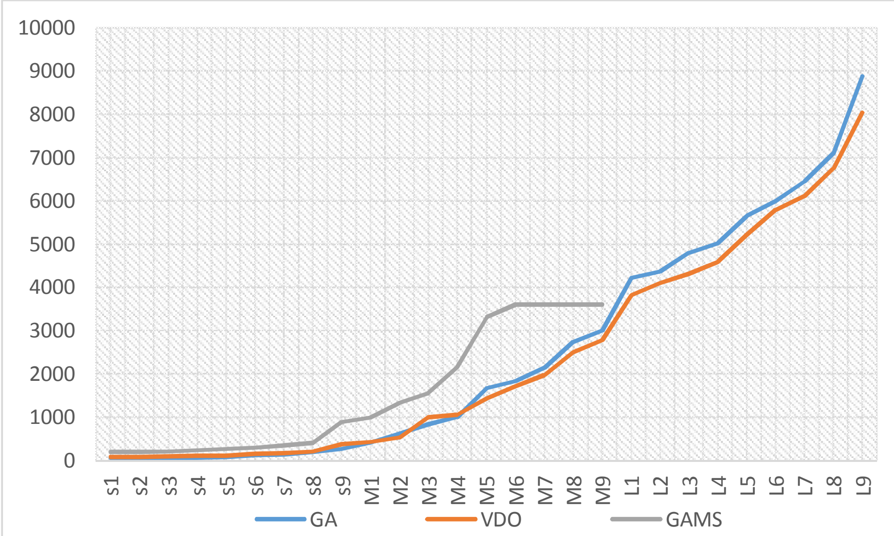


Fig. 11. Comparison table of performing duration of GA and VDO

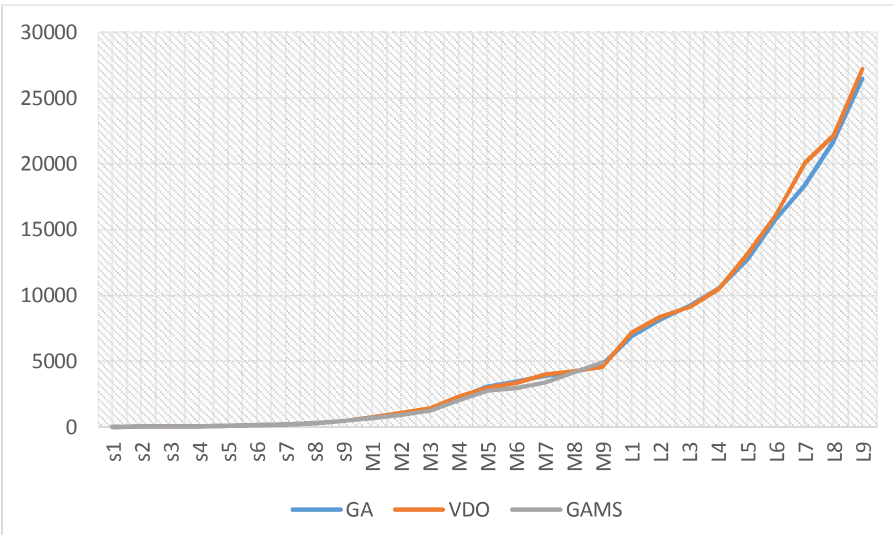


Fig. 12. Comparison of the results of GA and VDO

Examining Figure 11 and Table 5, it can be concluded that in large problems, the genetic algorithm provides better answers than the vibration damping algorithm, and in other problems, by comparing the answers from the two algorithms, neither is superior to the other. Also, by examining Figure 12 and Table 5, it can be concluded that, in small problems, the GA works faster than the VDO, and in large problems, the VDO is faster than the GA. To compare VDO, GA, and CPLEX algorithms, GAP and RPD criteria are used as (24) and (25) equations. To explain more briefly, relative percentage deviation (RPD) is the efficiency of the heuristics compared using the objective function values [48,49]. Also, problems have been investigated on small, medium, and large scales problems.

$$GAP=\frac{F_{metaheuristic}-F_{optimal}}{F_{optimal}}\times 100 \qquad (24)$$

$$RPD^{2}=\frac{F_{metaheuristic}-F_{best}}{F_{best}}\times 100 \qquad (25)$$

[2] Relative Percentage Deviation (RPD)

Tab. 6. Comparison table of GA and VDO proposed algorithm solutions and CPLEX solver solutions in small scale

Test Problem					GAMS (CPLEX Solver)		GA		GAP (GA-GAMS)	VDO		GAP(VDO-GAMS)
NO	n	m	u	p	C_{MAX} Average (in 5 runs)	Run Time(sec)	C_{MAX} Average (in 5 runs)	Run Time(sec)		C_{MAX} Average (in 5 runs)	Run Time(sec)	
SMALL Problem Size												
S1	3	3	2	2	50	198	50	52	0	50	90	0
S2	4	4	3	2	67	201	67	53	0	67	90	0
S3	5	4	4	2	81	216	81	61	0	81	97	0
S4	6	5	3	2	97	240	97	79	0	97	110	0
S5	6	5	4	2	113	263	113	87	0	113	112	0
S6	7	5	3	3	169	301	169	124	0	169	151	0
S7	8	5	4	3	202	345	202	137	0	202	169	0
S8	9	6	4	3	294	410	294	191	0	294	209	0
S9	9	7	6	3	486	893	486	265	0	486	379	0

Tab. 7. Comparison table of GA and VDO proposed algorithm solutions and CPLEX solver solutions in medium scale

Test Problem					GAMS (CPLEX Solver)		GA		GAP(GA-GAMS)	VDO		GAP(VDO -GAMS)	RPD(GA-VDO)
NO	n	m	u	p	C_{MAX} Average (in 5 runs)	Run Time(sec)	C_{MAX} Average (in 5 runs)	Run Time(sec)		C_{MAX} Average (in 5 runs)	Run Time(sec)		
MEDIUM Problem Size													
M1	11	9	7	4	721	986	762	416	5.6	745	441	3.3	2.3
M2	13	10	10	4	937	1342	1039	627	10.8	1106	525	18	6.4
M3	15	13	12	4	1246	1566	1351	833	8.4	1444	998	15.8	6.8
M4	17	15	15	4	2039	2164	2231	1001	9.4	2335	1054	14.5	4.6
M5	18	16	15	5	2781	3306	3081	1674	10.7	2984	1431	7.2	3.2
M6	19	16	15	5	3004	3600	3515	1832	17	3366	1710	12	4.4
M7	19	18	17	5	3420	3600	3929	2149	14.8	4023	1982	17.6	2.4
M8	20	18	17	5	4197	3600	4197	2743	0	4301	2509	2.4	2.4
M9	20	19	18	5	4949	3600	4791	3006	3.2	4580	2781	8	4.6

Tab. 8. Comparison table of GA and VDO proposed algorithm solutions and CPLEX solver solutions in large scale

Test Problem					GAMS (CPLEX Solver)		GA		VDO		RPD(GA -VDO)
NO	n	m	u	p	C_{MAX} Average (in 5 runs)	Run Time(sec)	C_{MAX} Average (in 5 runs)	Run Time(sec)	C_{MAX} Average (in 5 runs)	Run Time(sec)	
LARGE Problem Size											
L1	30	20	20	6	---	---	6941	4219	7210	3811	3.8
L2	35	25	25	7	---	---	8199	4371	8393	4099	2.3
L3	40	30	30	8	---	---	9214	4794	9140	4317	0
L4	45	35	35	9	---	---	10541	5016	10499	4588	9
L5	50	40	40	10	---	---	12737	5642	13121	5212	3
L6	55	45	45	11	---	---	15789	6001	16124	5791	2.1
L7	60	50	50	12	---	---	18354	6435	20019	6099	9
L8	65	55	55	13	---	---	21743	7091	22142	6749	2
L9	70	60	60	14	---	---	26456	8869	27187	8039	3

According to the GAP values in the small instances, GA, VDO, and GAMS obtain the same solutions; consequently, both metaheuristics can obtain optimal solutions for small-sized instances. Given the mean RPD values, the performance of GA as compared to VDO is better in the case of larger problem sizes. In other

problems, by comparing the answers from the two algorithms, neither is superior to the other.
To conclude which of the proposed algorithms has better performance, a statistical comparison is used. In statistical comparison, decision-making can be made using the obtained P-value. Then, for a statistically significant comparison between GAMS, GA, and VDO, analysis of variance (ANOVA) was used using Minitab 19 software. The statistical comparison for the objective function value is presented in Table 9 for small and medium problems and in Table 10 for large problems.

Tab. 9. ANOVA of the objective function of the small and medium problems

Source	DF	SS	MS	F	P-Value
Algorithms	2	94345	47173	0.02	0.983
Error	51	138435390	2714419		
Total	53	138529736			

Tab. 10. ANOVA of the objective function of the large

Source	DF	SS	MS	F	P-Value
Algorithms	1	828184	828184	0.02	0.895
Error	16	743755666	46484729		
Total	17	744583851			

The computational results in Tables 9 and 10 show that at 95% validity, there is no significant difference between the mean values of the objective function obtained by GAMS, GA, and VDO (P-values = 0.982>0.05, P-values = 0.895>0.05). Statistical comparisons for processing time are presented in small and medium problems in Table 11 and large problems in Table 12.

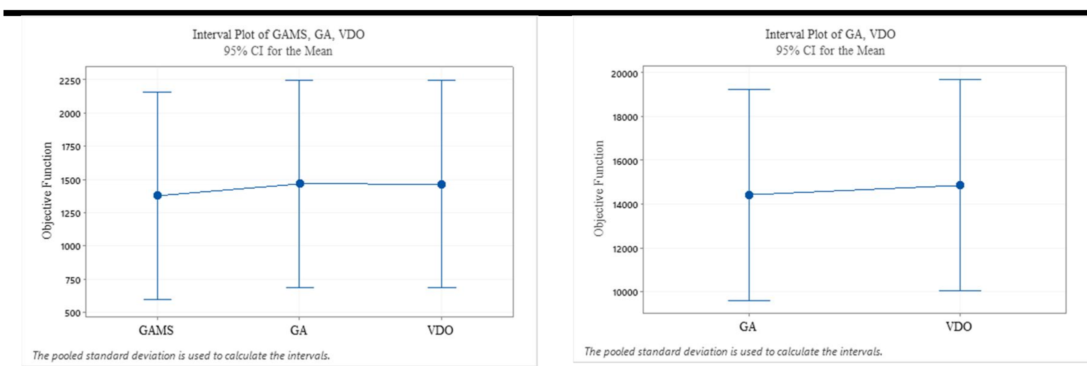
Tab. 11. Statistical comparisons for processing time in small and medium problems.

Source	DF	SS	Adj MS	F	P-Value
Algorithms	2	5117539	2558770	2.03	0.142
Error	51	64382104	1262394		
Total	53	69499643			

Tab. 12. Statistical comparisons for processing time large problems.

Source	DF	SS	MS	F	P-Value
Algorithms	1	774183	774183	0.37	0.550
Error	16	33225510	2076594		
Total	17	33999693			

The computational results in Tables 11 and 12 show that at 95% validity, there is no significant difference between the mean values of the objective function obtained by GAMS, GA, and VDO ((P-values=0.142, P-values = 0.550>0.05). Interval plots for the processing time and the value of the objective function are also presented in Figure 13.



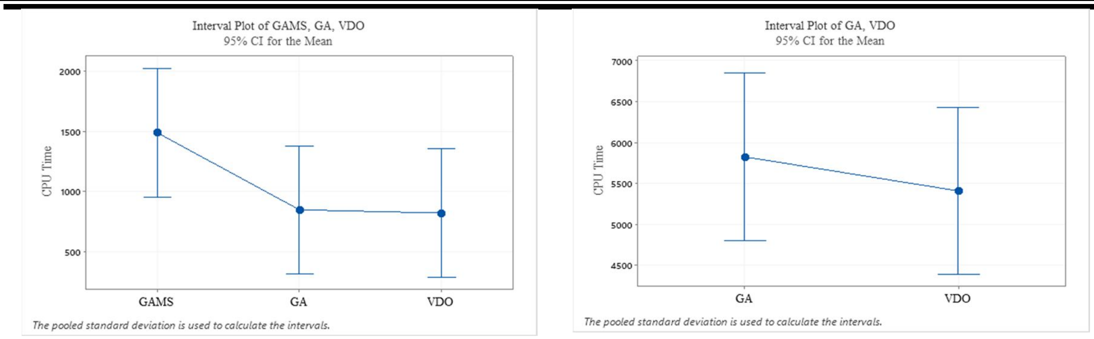


Fig. 13. Interval plots for the processing time and the objective function

The proposed approach of GA and VDO can provide an almost optimal solution (with a small distance from the optimal state) for small and medium-scale problems. Also, for solving large-scale problems, their solutions have a higher quality than CPLEX (it should be noted that CPLEX did not reach the optimal solution in 3600 seconds), so GA and VDO are suitable alternatives to CPLEX in solving large-scale DRCFJSP problems. If the size of the problem increases, the use of CPLEX requires an increase in the solution time much more than the proposed GA and VDO. Also, changes in tasks and machines have a greater impact on solution time.

6. Conclusion

In modern production systems, orders are processed using a set of resources, and in this regard, it is very important to develop a suitable program that optimally allocates resources to orders and determines the sequence of operations. With such planning, the last task completion time can be improved and the system remains close to optimal. In this paper, the double resource constraint flexible job-shop scheduling problem (DRCFJSP) considering the limit of preventive maintenance (PM) is examined. First, a mathematical model of the problem based on assumptions, variables, and constraints was presented to minimize the maximum makespan. In this regard, a mixed integer linear programming model (MILP) is presented for the problem, and to evaluate and validate the mathematical model, several small and medium examples are randomly generated using CPLEX solver in GAMS software. Since the problem is NP-hard, the genetic algorithm and the vibration damping optimization have been used to solve the problem on large scales. 27 problems with small, medium, and large dimensions were designed and solved and then a comparison was made between the answers obtained from GAMS software and the proposed algorithms as well as a comparison was made between two meta-heuristic algorithms. The computational results show that GAMS software can solve small problems in an acceptable time and achieve accurate answers, and meta-heuristic algorithms are also able to obtain appropriate answers in such a way that both algorithms spending a short time, find suitable answers in all aspects of the problem. Also, the efficiency of the two proposed algorithms are compared in terms of computational time and the value of the objective function. The computational results show that, for small problems at 95% validity, there is no significant difference between the mean values of the objective function obtained by GAMS, GA, and VDO. Also, for large problems at 95% validity, there is no significant difference between the mean values of the objective function obtained by GA and VDO. Interval plots for the processing time were also presented. Interval plots showed that in small dimensions, there is not much difference between the two metaheuristic algorithms in terms of computational time, but in large dimensions, the efficiency of the vibration damping algorithm is better. Comparing this research with other research, it can be seen that the problems are investigated from the point of view of the type of scheduling in the classical JSP and FJSP modes. This research is FJSP type. Also, from the point of view of resources, it can be seen that the problems are expressed in two modes, single resource or SRC and limited dual resources DRC. Also, this research is DRC type. In other words, it is the simultaneous consideration of limited human and machine resources with different job processing capabilities in an FJSP production system. From the perspective of dealing with the unexpected breakdown of machines and disruption in the workshop, this research considers preventive maintenance and repairs in machines. Another variation in this research is the application of a meta-heuristic solution

approach based on GA and VDO to solve the DRCFJSP problem in large dimensions. In the DRCFJSP problem of this research, unlike the majority of research in the field of FJSP modeling, which assumes that machines are always available and do not break down, we ignore these two almost unrealistic assumptions.
In future research, problem-solving using other meta-heuristics methods and comparison with the two proposed algorithms can be mentioned. Another suggestion can be to use the methods used in solving the optimization problems of combinations for the mathematical model presented in this article. The problem can also be solved by considering set-up times dependent on sequence and transportation. Or the problem can be solved by taking into account human error and the different skills of the workforce.]

References

[1] Pinedo, M., Scheduling. Springer, (2012).

[2] Pinedo, M., C. Zacharias, and N. Zhu, Scheduling in the service industries: An overview. Journal of systems science and systems engineering, Vol. 24, No. 1, (2015), pp. 1-48.

[3] Mehdizadeh, E., R. Tavakkoli-Moghaddam, and M. Yazdani, A vibration damping optimization algorithm for a parallel machines scheduling problem with sequence-independent family setup times. Applied Mathematical Modelling, Vol. 39, No. 22, (2015), pp. 6845-6859.

[4] Brucker, P. and R. Schlie, Job-shop scheduling with multi-purpose machines. Computing, Vol. 45, No. 4, (1990), pp. 369-375.

[5] Zhang, J. and J. Jie. A Multi-objective Particle Swarm Optimization Algorithm Embedded with Maximum Fitness Function for Dual-Resources Constrained Flexible Job Shop Scheduling. in International Conference on Intelligent Computing. Springe, (2021).

[6] Peng, C., et al. Analysis of double-resource flexible job shop scheduling problem based on genetic algorithm. in Networking, Sensing and Control (ICNSC), International Conference on. IEEE, (2018).

[7] Gong, G., et al., A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. Journal of Cleaner Production, Vol. 174, (2018), pp. 560-576.

[8] Garey, M.R., D.S. Johnson, and R .Sethi, The Complexity of Flowshop and Jobshop Scheduling. Mathematics of Operations Research, Vol. 1, No. 2, (1976), pp. 117-129.

[9] Zhang, X., J. Kang, and T. Jin, Degradation modeling and maintenance decisions based on Bayesian belief networks. IEEE Transactions on Reliability, Vol. 63, No. 2, (2014), pp. 620-633.

[10] Cha, J.H., M. Finkelstein, and G. Levitin, On preventive maintenance of systems with lifetimes dependent on a random shock process. Reliability Engineering & System Safety, Vol. 168, (2017), pp. 90-97.

[11] Han, Y., X .Zhen, and Y. Huang, Multi-objective optimization for preventive maintenance of offshore safety critical equipment integrating dynamic risk and maintenance cost. Ocean Engineering, Vol. 245, (2022), p. 110557.

[12] Gustavsson, E., et al., Preventive maintenance scheduling of multi-component systems with interval costs. Computers & Industrial Engineering, Vol. 76, (2014), pp. 390-400.

[13] Yang, L., et al., A two-phase preventive maintenance policy considering imperfect repair and postponed replacement. European Journal of Operational Research, Vol. 274, No. 3, (2019), pp. 966-977.

[14] Aggoune, R., Minimizing the makespan for the flow shop scheduling problem with availability constraints. European Journal of Operational Research, Vol. 153, No. 3, (2004), pp. 534-543.

[15] Gao, J., M. Gen, and L. Sun, Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. Journal of Intelligent Manufacturing, Vol. 17, No. 4, (2006), pp. 493-507.

[16] Naderi, B., M. Zandieh, and S.F. Ghomi, Scheduling sequence-dependent setup time job shops with preventive maintenance. The International Journal of Advanced Manufacturing Technology, Vol. 43, Nos. 1-2, (2009), p. 170.

[17] Mati, Y., Minimizing the makespan in the non-preemptive job-shop scheduling with limited machine availability. Computers & Industrial Engineering, Vol. 59, No. 4, (2010), pp. 537-543.

[18] Liu, X.X., C.B. Liu, and Z. Tao. Research on bi-objective scheduling of dual-resource constrained flexible job shop. in Advanced Materials Research. Trans Tech Publ, (2011).

[19] Dalfard, V.M. and G. Mohammadi, Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints. Computers & Mathematics with Applications, Vol. 64, No. 6, (2012), pp. 2111-2117.

[20] Kim, B.S. and Y. Ozturkoglu, Scheduling a single machine with multiple preventive maintenance activities and position-based deteriorations using genetic algorithms. The International Journal of Advanced Manufacturing Technology, (2013), pp. 1-11.

[21] Lei, D. and X. Guo, Variable neighbourhood search for dual-resource constrained flexible job shop scheduling. International Journal of Production Research, Vol. 52, No. 9, (2014), pp. 2519-2529.

[22] Yazdani, M., et al., Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem. Scientia Iranica. Transaction E, Industrial Engineering, Vol. 22, No. 3, (2015), p. 1242.

[23] Zheng, X.-l. and L. Wang, A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem. International Journal of Production Research, Vol. 54, No. 18, (2016), pp. 5554-5566.

[24] Paksi, A. and A. Ma'ruf. Flexible Job-Shop Scheduling with Dual-Resource Constraints to Minimize Tardiness Using Genetic Algorithm. in IOP Conference Series: Materials Science and Engineering. IOP Publishing, (2016).

[25] Gao, L. and Q.-K. Pan, A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem. Information Sciences, Vol. 372, (2016), pp. 655-676.

[26] A. Rahmati, S.H., A. Ahmadi, and B. Karimi, Developing Simulation Based Optimization Mechanism for Novel Stochastic Reliability Centered Maintenance Problem. Scientia Iranica, (2017).

[27] Zhang, J., et al., A hybrid particle swarm optimisation for multi-objective flexible job-shop scheduling problem with dual-resources constrained. International Journal of Computing Science and Mathematics, Vol. 8, No. 6, (2017), pp. 526-532.

[28] Cui, W.-W. and Z. Lu, Minimizing the makespan on a single machine with flexible maintenances and jobs' release dates. Computers & Operations Research, Vol. 80, (2017), pp. 11-22.

[29] Gong, X., et al., A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility. International Journal of Production Research, Vol. 56, No. 7, (2018), pp. 2506-2522.

[30] Yazdani, M., M. Zandieh, and R. Tavakkoli-Moghaddam, Evolutionary algorithms for multi-objective dual-resource constrained flexible job-shop scheduling problem. Opsearch, Vol. 56, No. 3, (2019), pp. 983-1006.

[31] Kress, D., D. Müller, and J. Nossack, A worker constrained flexible job shop scheduling problem with sequence-dependent setup times. OR Spectrum, Vol. 41, No. 1, (2019), pp. 179-217.

[32] Andrade-Pineda, J.L., et al., Scheduling a dual-resource flexible job shop with makespan and due date-related criteria. Annals of Operations Research, Vol. 291, No. 1, (2020), pp. 5-35.

[33] Soofi, P., et al., Robust Fuzzy-Stochastic Programming Model and Meta-heuristic Algorithms for Dual-Resource Constrained Flexible Job-Shop Scheduling Problem under Machine Breakdown. IEEE Access, (2021).

[34] Zhang, S., et al., Dual resource constrained flexible job shop scheduling based on improved quantum genetic algorithm. Machines, Vol. 9, No. 6, (2021), p. 108.

[35] Vital-Soto, A., M.F. Baki, and A. Azab, A multi-objective mathematical model and evolutionary algorithm for the dual-resource flexible job-shop scheduling problem with sequencing flexibility. Flexible Services and Manufacturing Journal, (2022), pp. 1-43.

[36] Hajibabaie, M. and J. Behnamian, Modeling and solving bi-objective flexible job shop scheduling with parallel machines and dual human-machine resources. Journal of Industrial Management Perspective, (2022).

[37] Victer Paul, P., et al., A novel ODV crossover operator-based genetic algorithms for traveling salesman problem. Soft Computing, Vol. 24, No. 17, (2020), pp. 12855-12885.

[38] Alizadeh ,R., et al., A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities. Computers & Industrial Engineering, Vol. 139, (2020), p. 106106.

[39] Luo, Q., et al., An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers. Expert Systems with Applications, Vol. 160, (2020), p. 113721.

[40] Wu, R., et al., Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. Advances in Mechanical Engineering, Vol. 10, No. 10, (2018), p. 1687814018804096.

[41] Abreu, L.R., R.F. Tavares-Neto, and M.S. Nagano, A new efficient biased random key genetic algorithm for open shop scheduling with routing by capacitated single vehicle and makespan minimization. Engineering Applications of Artificial Intelligence, Vol. 104, (2021), p. 104373.

[42] Wang, J.-J., Y.-Y. Jing, and C.-F. Zhang, Optimization of capacity and operation for CCHP system by genetic algorithm. Applied Energy, Vol. 87, No. 4, (2010), pp. 1325-1335.

[43] Yazdani, M., et al., Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem. Scientia Iranica, Vol. 22, No. 3, (2015), pp. 1242-1257.

[44] Yazdi, M.K. and R.T. Moghaddam, A multi-objective vibration damping meta-heuristic algorithm for multi-objective p-robust supply chain problem with travel time. Journal of Industrial and Systems Engineering, 11(Special issue: 14th International Industrial Engineering Conference): Vol. 11, (2018), pp. 176-189.

[45] Hajipour, V., E. Mehdizadeh, and R. Tavakkoli-Moghaddam, A novel Pareto-based multi-objective vibration damping optimization algorithm to solve multi-objective optimization problems. Scientia Iranica. Transaction E, Industrial Engineering, Vol. 2, No. 6, (2014), p. 2368.

[46] Mehdizadeh, E. and R. Tavakkoli-Moghaddam. Vibration damping optimization algorithm for an identical parallel machine scheduling problem. in Proceeding of the 2nd International Conference of Iranian Operations Research Society, Babolsar ,Iran. (2009).

[47] Aghajani-Delavar, N., et al., A multi-objective vibration damping optimization algorithm for solving a cellular manufacturing system with manpower and tool allocation. Scientia Iranica, (2020).

[48] Boosaiedi, S., M. Reisi-Nafchi, and G. Moslehi, Operating Room Scheduling Considering Patient Priorities and Operating Room Preferences: A Case Study. International Journal of Industiral Engineering & Producion Research, Vol. 33, No. 2, (2022), pp. 1-21.

[49] Mousavipour, S., H. Farughi, and F. Ahmadizar, A Job Shop Scheduling Problem

with Sequence-Dependent Setup Times Considering Position-Based Learning Effects and Availability Constraints. International Journal of Industiral Engineering & Producion Research, Vol. 30, No. 3, (2019), pp. 329-340.

Follow this article at the following site:

Amir Nayeb, Esmaeil Mehdizadeh & Seyed Habib A. Rahmati: A mathematical model for double resource constraint flexible job-shop scheduling problem considering the limit of preventive maintenance. IJIEPR 2023; 34 (2) :1-20
URL: http://ijiepr.iust.ac.ir/article-1-1561-en.html