Research Article GU J Sci 36(4): 1563-1576 (2023) DOI: 10.35378/gujs.1108155

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

A Hybrid Algorithm for Flow Shop Scheduling Problem with Unavailable Time Periods and Additional Resources

Feristah OZCELIK*, Tugba SARAC

Department of Industrial Engineering, Faculty of Engineering and Architecture, Eskişehir Osmangazi University, 26480, Eskisehir, Turkiye

Highlights

- Flow shop scheduling problem with unavailable periods and additional resources is discussed.
- Unavailable periods are flexible in a time window.
- A MIP model and a hybrid algorithm have been developed.
- In the proposed hybrid algorithm, genetic algorithm and modified subgradient algorithm works together.
- With the developed hybrid algorithm, GAMS results are improved up to 88%.

Article Info

Received: 24 Apr 2022
Accepted: 21 Dec 2022

Keywords

Flow shop scheduling
Unavailable periods
Additional resources
Hybrid algorithm

Abstract

In the scheduling literature, the studies that consider unavailable periods (UPs) have generally ignored the resources. However, when the resources to be used in unavailable periods are limited and these resources are needed for more than one machine at the same time, the problem of when the resource should be allocated to which machine arises. This decision is important as it can greatly affect the effectiveness of the machine schedule. For this reason, it is necessary to consider not only the UPs, but also the resources used by the UPs. In this study, flow shop scheduling problem with unavailable periods, flexible in a time window, and additional resources is discussed. In the considered problem, since additional resources are required during the unavailable periods and they can serve just one machine at a time, they cannot overlap. A MIP model and a hybrid algorithm that genetic algorithm and modified subgradient algorithm works together, have been developed for the considered problem. The performance of the hybrid algorithm is compared with pure genetic algorithm and Cplex solver of GAMS by using randomly generated test problems. Test results showed that while hybrid algorithm has solution quality advantage, genetic algorithm has solution time advantage. In addition, with the developed hybrid algorithm, GAMS results were improved up to 88%.

1. INTRODUCTION

Manufacturers must consider their production environment's conditions to obtain efficient schedules. Taking into consideration the maintenance activities and resource constraints in production scheduling is very critical in modern-day manufacturing and service environments [1]. Since handling these constraints increases the complexity of the problem, most studies in the scheduling literature, assume that machines are always available or ignore the additional resources. However, in many sectors, production is interrupted periodically due to planned maintenance and rest breaks. Furthermore, additional resources such as tools or a specialized workforce are required frequently in production or maintenance activities. For that reason, these constraints need to be considered when making scheduling decisions. However, as Geurtsen et al. [1] stated the number of studies that consider both unavailable periods (UP) and additional resources (AR) is still limited. The first study on machine scheduling problems with UP and AR was in 2000 [2]. The studies carried out from this date to the present are presented in Table 1.

*Corresponding author, e-mail: fdurmaz@ogu.edu.tr

Table 1. *Studies on machine scheduling with UP and AR*

Reference	*ME*	*TW*	*Objectives*	*Solution Method*
Lee and Chen [2]	P	✓	total weighted completion time	B&B
Yoo and Lee [3]	P	✓	multiple objective functions	DP
Belkaid et al. [4]	P		makespan	GA, MIP
Wong et al. [5]	P		makespan	GA
Wong et al. [6]	P		makespan	GA
Wang and Liu [7]	P		makespan	GA
Fu et al. [8]	P		makespan	PSO
Liu and Wang [9]	R		makespan, total completion time, resource cost	PTA
Rebai et al. [10]	R		total weighted completion time, total tardiness cost maintenance, total earliness cost maintenance	MIP, GA, H
Tavana et al. [11]	R		multiple objective functions	IP
Li et al. [12]	R		makespan	ABC-AC
Aramon Bajestani and Beck [13]	F		total maintenance and lost production costs	ITA
Boufellouh and Belkaid [14]	F		makespan and total production costs	NSGA-II, BOPSO
Wang and Yu [15]	J	✓	makespan, maximum machine workload, total machine workload	FBS
Fu et al. [16]	J		makespan	PSO
This study	F	✓	makespan	HA

ME: Machine Environment, P: identical parallel machines, R: unrelated parallel machines, F: Flow Shop, J: Job Shop, TW: time window , GA: Genetic Algorithm, HA: Hybrid Algorithm, B&B: branch and bound, PTA: Polynomial time algorithm, DP: Dynamic programming, H: Heuristic, MIP: Mixed Integer Programming, IP: Integer Programming, FBS: Filtered Beam Search, PSO: Particle Swarm Optimization, ABC-AC: Artificial Bee Colony with Adaptive Competition, NSGA-II: Non-Dominated Sorting Genetic Algorithm, BOPSO: Bi-Objective Adaptation of the Particle Swarm optimization, ITA: Integrated two-stage algorithm.

The addition of UP and AR constraints increases the complexity of scheduling problems. Therefore, as can be seen from Table 1, most of the studies dealing with these constraints together are in the parallel machine environment (P and R), which is less complex than flow shop (F) and job shop (J) scheduling problems.

Makespan is a manufacturer-oriented objective function as it serves to increase machine efficiency. Therefore, it cannot be ignored by many firms. Also, this objective function serves to balance machine loads in parallel machine scheduling problems. Therefore, it is one of the most frequently discussed objective functions in the scheduling literature. Similarly, as seen in Table 1, makespan is the most frequently used objective function in studies that UP and AR constraints are considered together.

Due to its NP-hard nature, it is not possible to solve the large sized of machine scheduling problems that deal with UP and AR constraints together with exact solution methods, so as can be seen from Table 1, metaheuristic algorithms are generally suggested as a solution method. Among the metaheuristic algorithms, GA has become one of the most preferred algorithms due to its success. GA is a method for solving both constrained and unconstrained optimization problems. The application of GAs to constrained optimization problems is often a challenging effort. Several methods such as repairing, penalizing, etc. have been proposed for handling constraints. The most common method in GAs to handle constraints is to use a penalty function. However, determining the value of the penalty parameter is also a difficult problem. If the value of it is not proper, even a feasible solution may not be obtained. Therefore, in this study, a hybrid algorithm that modified subgradient (MSG) algorithm and GA work together is proposed.

The MSG algorithm was proposed by Gasimov [17] for solving dual problems constructed in respect to a sharp augmented Lagrangean function. It is proven that the sharp augmented Lagrangean guarantees the zero-duality gap, if the objective and constraint functions are all Lipschitz [17]. Then, Kasimbeyli et al. [18] proposed the feasible value based modified subgradient (FMSG) algorithm as an improved version of the MSG algorithm. Detailed information about these algorithms can be reached from Gasimov [17] and Kasimbeyli et al. [18]. MSG and FMSG algorithms were utilized in the literature by many researchers to solve different nonconvex optimization problems such as quadratic assignment problem [19], quadratic knapsack problem [20], cell formation problem [21], multi-period facility layout problem [22], generalized quadratic multiple knapsack problem [23], capacitated vehicle routing problem [24] and aircraft maintenance routing problem [25]. Since the performances of the both MSG and FMSG algorithms depend on the performance of the solution method used for solving the sub problem at any iteration, some researchers hybridized these algorithms with metaheuristics. Ozcelik and Saraç [21] and Takan and Kasımbeyli [24] hybridized MSG with genetic algorithm, Saraç and Sipahioglu [23] hybridized FMSG with GA and Bulbul and Kasımbeyli [25] with ant colony optimization algorithm. When the hybrid algorithms in which MSG/FMSG algorithms and metaheuristic algorithms work together in the literature are examined, it has been observed that these studies have been applied to constrained problems, whose constraints are hard to handle with metaheuristics, and successful results have been obtained for these. Therefore, in this study, a hybrid algorithm in which MSG and GA work together is proposed for the FSS with UP and AR constraints for the first time.

Unavailable periods can be considered in two ways; flexible in a time window or fixed. Most of the studies in related literature assume the unavailable periods as fixed, that is, the starting time and the end time of UPs are pre-determined. However, in practice, since the starting time of UPs is generally flexible, availability constraints on the machine are non-fixed [15]. This means that the starting time of the UPs is not known in advance and has to be determined within the given time window during the scheduling process. The studies that considered the availability constraints as non-fixed are limited; such as Lee and Chen [2], Yoo and Lee [3] in identical parallel machine environment and Wang and Yu [15] in job shop environment. From the reached literature, there is not any study that considers UPs as flexible in a time window in FSS with UP and AR constraints.

Since FSS is an important problem commonly encountered in mass production systems, it has been frequently addressed by researchers in recent years. For detailed information on studies dealing with the FSS problem, see the review papers of Singh et al. [26], Komaki et al. [27], and Yenisey and Yağmahan [28].

There are only two studies considers FSS with UP and AR constraints. Aramon Bajestani and Beck [13] addressed an integrated maintenance and production scheduling problem. They solved the problem with an integrated two-stage algorithm. In the first stage, machines and time periods are assigned to maintenance, while in the second stage a schedule is created for the current time period. These two stage is solved iteratively until the solution costs in two stages converge. Boufellouh and Belkaid [14] considered production scheduling, maintenance planning and resource supply rate decisions in a permutation flow shop environment. The authors modelled the problem in a bi-objective manner, to minimize the makespan and total production costs, and proposed NSGA-II and BOPSO for the solution of the problem. None of these two studies considered the UP as flexible in a time window.

In this study, a flow shop scheduling problem with unavailable periods, flexible in a time window, and additional resource constraints is considered. To solve the problem, a hybrid algorithm that GA and MSG algorithm works together is proposed. Proposed solution method has a big advantage in dealing with complex constraints.

In the following section, the problem addressed is defined and the developed mathematical model is presented. The details of the proposed GA and hybrid algorithm (HA) are given in the third section, and the experimental results are presented in the fourth section. In the fifth section, the obtained results and suggestions for the future are discussed. In Appendix, we state the best-found sequences with GAMS, GA or HA.

2. PROBLEM DEFINITION AND DEVELOPED MATHEMATICAL MODEL

In the considered problem, n jobs are processed with the same order on m machines. The processing time (p_{jl}) of job *j* on each machine may be different. Job setup times are machine and sequence dependent. Jobs are ready at time zero. Each machine can only perform one job at a time. Each machine has available and unavailable periods. Available period *l* must occur within a certain interval. A job cannot be assigned to an unavailable period and cannot be split. Therefore, it must either be completed before or start after the unavailable period. During UPs, additional resources are used. ARs can serve only one machine at a time, in other words, there is a single server in the system. For that reason, it is desired that the UPs on different machines should not overlap. The objective function is to minimise the completion time of the last job on the last machine.

The proposed mathematical model (FSS-UP-AR) and the sets, indices, parameters, decision variables, constraints and objective function of this model are given below;

Sets and indexes:
N= {1, 2, .., n} set of job. $i, j, k \in N$
M = {1, 2, .., m} set of machines. $l, r \in M$
Q = {1, 2, .., $\max_i \delta_i$} set of number of UPs. $f \in Q$ where δ_i is the number of UPs in machine *i*

Parameters:
p_{jl}: processing time of job *j* in machine *l*
h_{jl}: setup time of job *j* in machine *l* (if *j* is the first job)
s_{ijl}: sequence and machine dependent setup time of job *j* in machine *l* after job *i*
θ: big enough positive number
b_l: duration of UP in machine *l*
σ_l^{min}: minimum time of available period in machine *l*
σ_l^{max}: maximum time of available period in machine *l*

Decision variables:
x_{jk}: 1, if job *j* is assigned k^{th} sequence; 0, otherwise.
C_{kl}: completion time of job *j* assigned k^{th} sequence in machine *l*
a_{kl}: starting time of job *j* assigned k^{th} sequence in machine *l*
a_{fl}^{UP}: starting time of UP *f* in machine *l*
C_{fl}^{UP}: completion time of UP *f* in machine *l*
w_{kfl}^{S}: 1, if job in k^{th} sequence at machine *l* is completed before UP *f* starts in the same machine; 0, if UP *f* is completed before job in k^{th} sequence at machine *l* starts in the same machine.
w_{flr}^{D}: 1, if UP *f* in machine *l* is completed before UP *f* is started in machine *r*; 0, if UP *f* in machine *r* is completed before UP *f* is started in machine *l*.

(FSS-UP-AR)
Objective function:

$$min\ z = C_{nm} \tag{1}$$

Constraints:

$$\sum_k x_{jk} = 1 \qquad \forall j \in N \tag{2}$$

$$\sum_j x_{jk} = 1 \qquad \forall\, k \in N \tag{3}$$

$$C_{11} = \sum_j (h_{j1} + p_{j1})\, x_{jk} \tag{4}$$

$$C_{1l} \geq C_{1(l-1)} + h_{jl} + p_{jl} - \theta\big(1 - x_{j1}\big) \qquad l > 1, \forall\, j \in N \tag{5}$$

$$C_{k1} \geq C_{(k-1)1} + p_{j1} + s_{ij1} - \theta\big(2 - x_{i(k-1)} - x_{jk}\big) \qquad k > 1, i \neq j \tag{6}$$

$$a_{kl} \geq C_{(k-1)l} \qquad l > 1, k > 1 \tag{7}$$

$$a_{kl} \geq C_{k(l-1)} \qquad l > 1, k > 1 \tag{8}$$

$$C_{kl} \geq a_{kl} + p_{jl} + s_{ijl} - \theta\big(2 - x_{i(k-1)} - x_{jk}\big) \qquad k > 1, l > 1, i \neq j \tag{9}$$

$$a_{11} = 0 \tag{10}$$

$$a_{1l} = C_{1l} - \sum_j (h_{jl} + p_{jl})\, x_{j1} \qquad l > 1 \tag{11}$$

$$a_{k1} \geq C_{k1} - p_{j1} - s_{ij1} - \theta\big(2 - x_{i(k-1)} - x_{jk}\big) \qquad k > 1, i \neq j \tag{12}$$

$$a_{k1} \leq C_{k1} - p_{j1} - s_{ij1} + \theta\big(2 - x_{i(k-1)} - x_{jk}\big) \qquad k > 1, i \neq j \tag{13}$$

$$a_{1l}^{UP} \geq \sigma_l^{min} \qquad \forall\, l \in M \tag{14}$$

$$a_{1l}^{UP} \leq \sigma_l^{max} \qquad \forall\, l \in M \tag{15}$$

$$C_{fl}^{UP} \geq C_{f-1,l}^{UP} + b_l + \sigma_l^{min} \qquad f > 1, \forall\, l \in M \tag{16}$$

$$C_{fl}^{UP} \leq C_{f-1,l}^{UP} + b_l + \sigma_l^{max} \qquad f > 1, \forall\, l \in M \tag{17}$$

$$a_{fl}^{UP} = C_{fl}^{UP} - b_l \qquad \forall\, f \in Q, \forall\, l \in M \tag{18}$$

$$C_{fl}^{UP} \leq a_{kl} + \theta\, w_{kfl}^{S} \qquad \forall\, k \in N, \forall\, f \in Q, \forall l \in M \tag{19}$$

$$C_{kl} \leq a_{fl}^{UP} + \theta\big(1 - w_{kfl}^{S}\big) \qquad \forall\, k \in N, \forall\, f \in Q, \forall l \in M \tag{20}$$

$$C_{fl}^{UP} \leq a_{fr}^{UP} + \beta\, w_{flr}^{D} \qquad r \neq l, \forall\, f \in Q \tag{21}$$

$$C_{fr}^{UP} \leq a_{fl}^{UP} + \beta\, \big(1 - w_{flr}^{D}\big) \qquad r \neq l, \forall\, f \in Q \tag{22}$$

$$x_{jk} \in \{0,1\} \qquad \forall\, j, k \in N \tag{23}$$

$$a_{kl}, C_{kl} \geq 0 \qquad \forall\, k \in N, \forall\, l \in M \tag{24}$$

$$a_{fl}^{UP}, C_{fl}^{UP} \geq 0 \qquad \forall\, f \in Q, \forall\, l \in M \tag{25}$$

$$w_{kfl}^{S} \in \{0,1\} \qquad \forall\, k \in N, \forall\, f \in Q, \forall l \in M \tag{26}$$

$$w_{flr}^{D} \in \{0,1\} \qquad \forall\, f \in Q, \forall l, r \in M \tag{27}$$

The objective (1) is to minimise the last job of last machine. Equations (2) and (3) are the assignment constraints. Equations (4)-(13) is for the calculation of the completion and starting times of the jobs and Equations (14)-(18) is also for the UPs. Equations (19)-(22) are the nonoverlapping constraints. Equations (19) and (20) avoids that the jobs are overlap with UPs in the same machine and Equations (21) and (22) avoids overlapping the UPs in the different machines. Equations (23)-(27) defines the types of decision variables.

3. PROPOSED SOLUTION METHODS

In this section, the proposed GA and the hybrid algorithm are presented in detail.

3.1. Genetic Algorithm

Genetic algorithm (GA) is a metaheuristic that mimics the evolutionary process in biology. A GA is an iterative procedure that searches the decision space at multiple points simultaneously. Here, each solution is represented by a chromosome, and chromosomes construct a population. For each generation, GA searches for better solutions using selection, crossover, and mutation operators. Steps of the proposed genetic algorithm are given below.

Steps of the genetic algorithm:

Initialization Step. Get the algorithm parameters. Generate and evaluate the initial population.
repeat
    selection
    crossover
    mutation
    evaluation
    elitizm
until the stopping criteria is satisfied

In the *Initialization Step*, firstly the values of GA parameters are read. In this study, the values of these parameters are taken as 50 for population size (*ps*), 0.6 for crossover rate (*cr*), and 0.1 for mutation rate (*mr*). Then, initial population is generated randomly. Each chromosome in the population represents to a solution (sequence of jobs). Permutation encoding structure is used as solution representation. A sample chromosome (2 3 4 5 1) shows that the five jobs will be processed in sequence 2, 3, 4, 5, and 1.

In the proposed GA, *selection*, *crossover*, *mutation*, *evaluation*, and *elitizm* steps are repeated until the best solution has not been able to improve for 10000 generations. In the *selection, crossover* and *mutation* steps, two-tournament, OX and insertion operators are preferred, respectively, since these operators have been successfully applied in the studies using permutation encoding structure [29].

Since the considered problem has constraints, in the *evaluation* step, firstly, the constraint handling method is determined. With the used solution representation, the assignment (2)-(3) constraints are guaranteed to be satisfied. The constraints of calculating the starting and completion times of jobs and UPs (4)-(20) do not require any extra effort to be provided. They are easily calculated with the solution of the corresponding chromosome. However, to satisfy the nonoverlapping constraint group (21)-(22) a constraint handling method is required. For this, the penalty method is used.

Fitness function ($fitness_{GA}$) of the proposed GA consist of two parts; the objective function of the proposed mathematical model (FSS- UP-AR) and penalty value. It is given in Equation (28)

$$fitness_{GA} = C_{nm} + \varphi \sum_{l} \sum_{r} \sum_{f} g_{lrf} \tag{28}$$

where φ is the penalty coefficient and is calculated using the formula $\varphi = 100nm$. Here, g_{lrf} is the amount of nonoverlapping constraint violations, calculated using Equation (29). For the satisfied constraints, the value of g_{lrf} is zero

$$g_{lrf} = \begin{cases} C_{fl}^{UP} - a_{fr}^{UP}; & a_{fr}^{UP} \le C_{fl}^{UP} \le C_{fr}^{UP} \\ C_{fr}^{UP} - a_{fl}^{UP}; & a_{fl}^{UP} \le C_{fr}^{UP} \le C_{fl}^{UP} \\ 0; & otherwise\ . \end{cases} \tag{29}$$

In the *elitizm* step, by replacing the best individual of the previous population with the worst individual of the current population, it is guaranteed that the successful solutions obtained in the former generations will be transferred to the new generations.

3.2. Hybrid Algorithm

In this paper, a hybrid algorithm, in which the nonoverlapping constraint group is provided by the MSG algorithm and the sub-problem of the MSG algorithm is solved with GA, is developed. The definitions and the steps of the hybrid algorithm are given below.

k : iteration number.
μ : number of constraints.
u^k and c^k : dual variables calculated at the k^{th} iteration.
$f(x)$: objective function of the problem
$g(x)$: constraint functions of the problem.
$\overline{H}$: upper bound for the dual problem
s^k: positive step size parameter.
α and δ: positive step size parameters.

Steps of the hybrid algorithm:

Initialization Step. Let k = 1. Choose a vector $(u^1, c^1) \in R^\mu \times R_+$ and the scalars $\alpha \in R_+$ and δ, a real number in range (0, 2).

Step 1. Solve the following sub-problem (SP-MSG) with given (u^k, c^k).
(SP-MSG):

$$\min_{x^k \in S} \; L(x^k, u^k, c^k) = f(x^k) + c^k \|g(x^k)\| - \langle u^k, g(x^k) \rangle \tag{30}$$

$$subject\ to\ f(x^k) + c^k \|g(x^k)\| - \langle u^k, g(x^k) \rangle \leq \overline{H} \tag{31}$$

where x^k belongs to the set S, $\|.\|$ is the Euclidean norm and <.,.> is the Euclidean inner product on R^μ.
If $\|g(x^k)\| = 0$ then *STOP*. (u^k, c^k), is a solution to the dual problem, x^k is a solution to primal problem. Otherwise, go to *Step 2*.

Step 2. $k = k + 1$. Update the s^k, u^k and c^k using the formulas given in the Equations (32), (33) and (34) respectively.

$$s^k = \frac{\delta\alpha\left(\overline{H} - L(x^k, u^k, c^k)\right)}{(\alpha^2 + (1+\alpha)^2)\left\|g(x^k)\right\|^2} \tag{32}$$

$$u^k = u^{k-1} - \alpha s^k g(x^k) \tag{33}$$

$$c^k = c^{k-1} + (1+\alpha) s^k \left\|g(x^k)\right\| \tag{34}$$

Go to *Step 1.*

In the hybrid algorithm, GA-MSG is used to solve the sub-problem of the MSG algorithm (SP-MSG). GA-MSG is the same as the GA, given in section 3.1, except for the fitness function. The fitness function of GA-MSG corresponds to the Lagrange function of the MSG algorithm and its formula is given in Equation (35)

$$fitness_{GA-MSG} = C_{nm} + c^k \sqrt{\sum_l \sum_r \sum_f g_{lrf}^2} - \sum_l \sum_r \sum_f u_{lrf}^k g_{lrf}. \tag{35}$$

The initial values of the c^0 and u^0_{lrf} parameters are taken as zero. As similar to GA, the population size (*ps*) of the developed HA is taken as 50, the crossover rate (*cr*) is 0.6, and the mutation rate (*mr*) is 0.1. If the solutions have not been able to be improved for 20 generations, HA is terminated.

4. COMPUTATIONAL RESULTS

All computations are performed on a 2.70 GHz i7 PC with 8 Gb of RAM. GAMS 24.0.2 Cplex solver is used to solve the proposed mathematical model. The time limit is 10800 seconds for GAMS/Cplex. The developed GA is coded in Borland Delphi 7.0. In section 4.1, the features of the test problems and in section 4.2, the obtained test results are given.

4.1. Test Problems

The performance of the proposed methods is tested by using randomly generated test problems. 12 problem types are generated, including six job-machine (*n*-*m*) combinations and two types of Δ values. The considered job-machine combinations are 10-2, 20-3, 30-4, 50-4, 100-5, 150-6 and Δ type takes the value 1, if $\Delta = \frac{1}{4}$ or 2, if $\Delta = \frac{1}{3}$. Processing times (p_{jl}) are generated from a discrete uniform distribution with a range between 20 and 100. Setup times (h_{jl}, s_{jkl}) and duration of UPs (b_l) are generated from discrete uniform distribution of [5, 20] and [20,30], respectively. Maximum time of available periods (σ_l^{max}) are generated by using the formula given in Equation (36)

$$\alpha_l^{max} = \delta n \Delta \qquad \text{where } \Delta \in \left\{\frac{1}{4}, \frac{1}{3}\right\}, \delta \sim U\{120,150\}. \tag{36}$$

Minimum time of available periods (α_l^{min}) are calculated by rounding 0.8 times α_l^{max} to the nearest integer number.

A total of 36 test problems are generated, three samples for each problem type. The test problems are named as *n*-*m*-Δ *type*-*sample number*. The test problems are available for download at https://drive.google.com/drive/folders/1udTmFc9iQYi6iTDPHQkeMGLRWjOZx6wN?usp=sharing.

4.2. Test Results

Generated test problems are solved with GAMS, GA and HA. The results of $n \leq 30$ test problems are presented in Table 2 and those with $n \geq 50$ in Table 3. The tables consist of five parts. Problem names are given in the first part, objective function values and CPU times of the GAMS, GA and HA are given in the second, the third and the fourth parts, respectively. The relative percentage improvement values between two solution methods (GAMS-GA, GAMS-HA, and GA-HA) are calculated and given in the last part of the tables. For GAMS-GA, the percentage improvement obtained by GA with respect to GAMS is calculated using the formula (37). The percentage improvement values of GAMS-HA and GA-HA are calculated as similar to GAMS-GA

$$imp_{GAMS-GA} = \frac{z_{GAMS} - z_{GA}}{z_{GAMS}} 100. \tag{37}$$

As seen from Table 2 and 3, with the GAMS/Cplex, for all of the 10 job problems optimal solutions are obtained within the time limit. While feasible solutions are obtained for 20, 30, and 50 job problems, they could not be found for $n \geq 100$.

Table 2. *Test results for $n \leq 30$ problems*

problem	GAMS		GA		HA		$imp_{GAMS-GA}$	$imp_{GAMS-HA}$	imp_{GA-HA}
	z	*t (sn.)*	*z*	*t (sn.)*	*z*	*t (sn.)*			
10-2-1-1	659	6543	659	<1	659	<1	0,00	0,00	0,00
10-2-1-2	854	7500	854	<1	854	<1	0,00	0,00	0,00
10-2-1-3	688	6167	691	<1	691	<1	-0,44	-0,44	0,00
10-2-2-1	803	11172	807	<1	807	<1	-0,50	-0,50	0,00
10-2-2-2	838	12588	838	<1	838	<1	0,00	0,00	0,00
10-2-2-3	717	8713	717	<1	717	<1	0,00	0,00	0,00
20-3-1-1	1627	10800	1610	3	1612	1	1,04	0,92	-0,12
20-3-1-2	1484	10800	1479	1	1476	1	0,34	0,54	0,20
20-3-1-3	1687	10800	1679	3	1679	1	0,47	0,47	0,00
20-3-2-1	1682	10800	1649	2	1646	1	1,96	2,14	0,18
20-3-2-2	1658	10800	1650	1	1649	1	0,48	0,54	0,06
20-3-2-3	1561	10800	1554	1	1548	1	0,45	0,83	0,39
30-4-1-1	2519	10800	2476	5	2464	19	1,71	2,18	0,48
30-4-1-2	2346	10800	2327	1	2320	13	0,81	1,11	0,30
30-4-1-3	2329	10800	2335	7	2325	13	-0,26	0,17	0,43
30-4-2-1	2412	10800	2400	3	2397	25	0,50	0,62	0,13
30-4-2-2	2483	10800	2475	2	2473	17	0,32	0,40	0,08
30-4-2-3	2503	10800	2472	9	2468	25	1,24	1,40	0,16

Table 3. *Test results for $n \geq 50$ problems*

problem	GAMS		GA		HA		imp_{GA-HA}
	z	t (sn.)	z	t (sn.)	z	t (sn.)	
50-4-1-1	29728	10800	3834	12	3817	17	0,44
50-4-1-2	4334	10800	3774	23	3758	30	0,42
50-4-1-3	25676	10800	3803	18	3785	20	0,47
50-4-2-1	4019	10800	3785	13	3719	27	1,74
50-4-2-2	4146	10800	3796	43	3760	19	0,95
50-4-2-3	30534	10800	3701	20	3689	23	0,32
100-5-1-1	-	10800	7686	36	7668	71	0,23
100-5-1-2	-	10800	7583	23	7554	135	0,38
100-5-1-3	-	10800	7854	56	7797	83	0,73
100-5-2-1	-	10800	7796	27	7777	119	0,24
100-5-2-2	-	10800	8032	60	8019	151	0,16
100-5-2-3	-	10800	7703	144	7607	137	1,25
150-6-1-1	-	10800	12118	58	11839	256	2,30
150-6-1-2	-	10800	12354	28	12056	221	2,41
150-6-1-3	-	10800	12343	109	12218	186	1,01
150-6-2-1	-	10800	-		12308	2600	-
150-6-2-2	-	10800	-		12310	3122	-
150-6-2-3	-	10800	-		12048	148	-

The proposed GA and HA reached the optimal solutions for four of six 10 job problems. They found better solutions than the GAMS for the remaining problems. When the results of 50 job problems, which are the largest problems GAMS can solve, are focused, it seems that the improvement rates are up to 88%.

Although pure GA, which uses the penalty method to ensure the nonoverlapping constraints, has a solution time advantage to HA, it could not find a feasible solution to the 150-6-2-1, 150-6-2-2, 150-6-2-3 problems. On the other hand, the proposed HA is both able to find feasible solutions for all test problems and improved the solutions of GA.

In Tables A1 and A2, we state the best-found sequences with GAMS, GA or HA for $n \leq 50$ problems and $n \geq 100$ problems, respectively.

5. CONCLUSIONS

In this study, flow shop scheduling problem with UP and AR constraints is discussed. A MIP model is proposed for the considered problem. Since it is not possible to solve problems larger than 50 jobs with the proposed mathematical model, an algorithm that can solve large-sized problems is needed. It is difficult to satisfy all of the UP and AR constraints by a metaheuristic. Including some of these constraints in the objective function with the help of the Lagrange function and satisfying them with the Subgradient algorithm may contribute to the performance of metaheuristic algorithms. In this study, firstly a pure GA, and then an HA, where AR constraints are included in the Lagrange function and the problem is solved with the MSG algorithm, are proposed. In the HA algorithm, the sub-problem of the MSG algorithm is solved with a GA that takes into account the constraints other than AR. The performance of the proposed solution methods is tested by using randomly generated test problems. With the developed algorithms, GAMS results have been improved up to 88%. Test results showed that while HA can obtain more successful solutions than GA, GA has a shorter solution time than HA. In addition, while HA can find feasible solutions for all problems, it has been observed that pure GA, using a penalty function to satisfy AR constraints, has difficulty in finding feasible solutions for $n \geq 150$ job problems. In the future, the effect of the hybridization of the MSG algorithm with other metaheuristics or the inclusion of different constraint groups in the Lagrangian function can be examined.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Geurtsen, M., Didden, J.B.H.C., Adan, J., Atan, Z., Adan, I., “Production, maintenance and resource scheduling: A review”, European Journal of Operational Research, (2022) (In press).

[2] Lee, C., Chen, Z., “Scheduling jobs and maintenance activities on parallel machines”, Naval Research Logistics, 47: 145–165, (2000).

[3] Yoo, J., Lee, I., “Parallel machine scheduling with maintenance activities”, Computers & Industrial Engineering, 101: 361–371, (2016).

[4] Belkaid, F., Dahane, M., Sair, Z., Khatab, A., “Efficient approach for joint maintenance planning and production scheduling under consumable resources constraints”, 44th International conference on computers& industrial engineering, (2014).

[5] Wong, C.S., Chan, F.T.S., Chung, S.H., “A genetic algorithm approach for production scheduling with mould maintenance consideration”, International Journal of Production Research, 50(20): 5683–5697, (2012).

[6] Wong, C., Chan, F.T., Chung, S., “Decision-making on multi-mould maintenance in production scheduling”, International Journal of Production Research, 52(19): 5640–5655, (2014).

[7] Wang, S., Liu, M., “Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning”, Journal of Manufacturing Systems, 37(1): 182–192, (2015).

[8] Fu, X., Chan, F.T., Niu, B., Chung, N.S., Qu, T., “A three-level particle swarm optimization with variable neighbourhood search algorithm for the production scheduling problem with mould maintenance”, Swarm and Evolutionary Computation, 50: 100572, (2019).

[9] Liu, C.L., Wang, J.J., “Unrelated parallel-machine scheduling with controllable processing times and impact of deteriorating maintenance activities under consideration”, Asia-Pacific Journal of Operational Research, 33(1): 1–16, (2016).

[10] Rebai, M., Kacem, I., Adjallah, K.H., “Scheduling jobs and maintenance activities on parallel machines”, Operational Research, 13(3): 363–383, (2013).

[11] Tavana, M., Zarook, Y., Santos-Arteaga, F.J., “An integrated three-stage maintenance scheduling model for unrelated parallel machines with aging effect and multi-maintenance activities”, Computers and Industrial Engineering, 83: 226–236, (2015).

[12] Li, M.B., Xiong, H., Lei, D.M., “An artificial bee colony with adaptive competition for the unrelated parallel machine scheduling problem with additional resources and maintenance”, Symmetry, 14(7): 1380, (2022).

[13] Aramon Bajestani, M., Beck, J.C., “A two-stage coupled algorithm for an integrated maintenance planning and flowshop scheduling problem with deteriorating machines”, Journal of Scheduling, 18(5): 471–486, (2015).

[14] Boufellouh, R., Belkaid, F., “Bi-objective optimization algorithms for joint production and maintenance scheduling under a global resource constraint: Application to the permutation flow shop problem”, Computers & Operations Research, 122: 104943, (2020).

[15] Wang, S., Yu, J., “An effective heuristic for flexible job-shop scheduling problem with maintenance activities”, Computers & Industrial Engineering, 59(3): 436–447, (2010).

[16] Fu, X., Chan, F.T., Niu, B., Chung, S.H., Bi, Y., “Minimization of makespan through jointly scheduling strategy in production system with mould maintenance consideration”, International Conference on Intelligent Computing, 577–586, Springer, (2017).

[17] Gasimov, R.N., “Augmented Lagrangian duality and nondifferentiable optimization methods in nonconvex programming”, Journal of Global Optimization, 24(2): 187-203, (2002).

[18] Kasimbeyli, R., Ustun, O., Rubinov, A.M., “The modified subgradient algorithm based on feasible values”, Optimization, 58(5): 535-560, (2009).

[19] Gasimov, R.N., Ustun, O., “Solving the quadratic assignment problem using F-MSG algorithm”, Journal of Industrial and Management Optimization, 3(2): 173-191, (2007).

[20] Sipahioglu, A, Saraç, T, “The performance of the modified subgradient algorithm on solving the 0-1 quadratic knapsack problem”, Informatica, 20(2): 1-12, (2009).

[21] Ozcelik, F., Saraç, T., “A genetic algorithm extended modified sub-gradient algorithm for cell formation problem with alternative routings”, International Journal of Production Research, 50(15): 4025-4037, (2012).

[22] Ulutas, B., Saraç, T., “Determining the parameters of MSG algorithm for multi period layout problem”, Journal of Manufacturing Technology Management, 7: 922–936, (2012).

[23] Saraç, T., Sipahioglu, A., “Generalized quadratic multiple knapsack problem and two solution approaches”, Computers & Operations Research, 43: 78-89, (2014).

[24] Takan, M.A., Kasımbeyli, R., “A hybrid subgradient method for solving the capacitated vehicle routing problem”, Journal of Nonlinear and Convex Analysis, 21(2): 413-423, (2020).

[25] Bulbul, K.G., Kasimbeyli, R., “Augmented Lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem”, Computers & Operations Research, 132: 105294, (2021).

[26] Singh, H., Oberoi, J.S., Singh, D., “Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: a systematic literature review”, Rairo-Operations Research, 55(1): 27-50, (2021).

[27] Komaki, G.M., Sheikh, S., Malakooti, B., “Flow shop scheduling problems with assembly operations: a review and new trends”, International Journal of Production Research, 57(10): 2926-2955, (2019).

[28] Yenisey, M.M., Yagmahan, B., “Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends”, OMEGA-International Journal of Management Science, 45: 119-135, (2014).

[29] Ozcelik, F., Islier, A.A., “Generalisation of unidirectional loop layout problem and solution by a genetic algorithm”, International Journal of Production Research, 49(3): 747-764, (2011).

Table A1. *Best-found sequences for $n \leq 50$ problems*

problem	*best ofv*	*best solution*
10-2-1-1	659	6 4 9 3 10 1 5 7 8 2
10-2-1-2	854	3 5 2 8 1 6 9 10 7 4
10-2-1-3	688	1 2 5 8 9 10 4 7 6 3
10-2-2-1	803	9 6 5 8 10 7 3 4 1 2
10-2-2-2	838	7 4 3 8 10 2 6 9 5 1
10-2-2-3	717	10 5 7 8 4 6 1 3 9 2
20-3-1-1	1610	2 8 16 7 11 10 5 6 15 1 17 19 9 13 4 12 18 20 14 3
20-3-1-2	1476	1 10 13 9 3 16 2 17 19 12 6 18 20 15 8 7 11 14 5 4
20-3-1-3	1679	20 6 8 17 5 19 1 16 18 11 2 12 4 15 9 13 3 14 7 10
20-3-2-1	1646	4 10 13 14 7 17 19 9 15 16 6 18 5 3 8 20 2 12 11 1
20-3-2-2	1649	7 11 16 18 4 19 17 6 8 12 15 2 13 20 5 9 1 14 10 3
20-3-2-3	1548	10 5 6 15 4 16 8 20 7 3 13 14 2 19 12 17 18 11 9 1
30-4-1-1	2464	6 5 1 30 22 19 17 3 13 4 29 7 8 24 18 28 11 10 14 20 21 12 15 25 26 23 2 9 27 16
30-4-1-2	2320	24 30 10 19 6 25 12 22 5 9 3 11 29 1 2 14 8 4 17 23 27 21 18 13 26 7 16 15 28 20
30-4-1-3	2325	19 27 3 12 13 22 17 21 7 18 4 14 23 1 16 24 28 26 29 9 11 6 30 10 5 25 2 15 20 8
30-4-2-1	2397	10 22 6 29 1 9 2 3 30 25 13 14 12 11 20 27 19 26 21 23 18 5 4 24 15 17 28 7 16 8
30-4-2-2	2473	5 7 2 17 25 14 30 29 18 23 22 3 1 20 12 27 19 8 11 26 4 21 6 24 9 13 10 15 16 28
30-4-2-3	2468	8 12 18 30 4 3 27 14 20 22 7 25 6 9 19 29 10 1 16 23 2 5 24 11 17 28 26 13 21 15
50-4-1-1	3817	13 14 49 6 19 18 42 43 17 47 15 16 44 31 24 39 3 20 25 2 27 12 35 5 11 22 1 40 48 28 7 4 10 41 32 33 29 34 37 23 30 8 9 38 46 45 26 50 21 36
50-4-1-2	3758	30 33 32 7 23 42 18 40 19 2 24 43 49 38 8 17 50 14 35 11 4 37 16 5 13 26 3 41 10 45 6 48 46 1 9 15 22 34 25 20 28 29 39 36 27 44 21 12 47 31
50-4-1-3	3785	15 23 13 4 49 24 30 39 8 46 47 17 32 48 2 1 37 16 7 34 21 35 29 33 45 12 31 38 44 5 36 11 40 3 14 42 26 22 28 10 43 18 27 19 41 6 50 25 9 20
50-4-2-1	3719	4 12 38 47 44 27 16 42 13 9 46 41 6 21 36 23 50 22 43 49 14 8 18 1 25 2 15 3 26 37 17 11 39 7 33 20 40 30 24 29 5 19 48 34 10 28 35 45 32 31
50-4-2-2	3760	15 8 5 50 21 11 6 17 43 34 2 31 35 39 9 22 33 28 30 7 1 46 14 48 41 3 44 26 19 32 16 36 20 25 4 27 42 37 47 13 40 45 38 18 49 24 12 10 29 23
50-4-2-3	3689	44 13 31 16 1 29 10 25 47 45 37 18 26 21 40 20 50 5 23 36 4 12 19 22 14 24 41 46 35 17 27 3 11 49 33 43 6 42 38 32 8 28 2 39 7 9 15 30 34 48

Table A2. *Best-found sequences for* $n \geq 100$ *problems*

problem	*best ofv*	*best solution*
100-5-1-1	7668	23 30 45 26 27 50 31 39 96 66 84 10 87 19 72 13 18 40 67 68 41 44 59 29 48 36 74 58 62 54 16 38 2 52 7 11 15 49 14 70 21 35 75 6 17 37 1 73 3 100 9 61 56 93 55 51 24 8 89 46 90 43 65 53 97 98 85 82 76 47 64 22 99 88 80 4 57 79 94 92 81 42 95 78 5 91 69 60 77 63 28 12 33 86 83 71 34 20 32 25
100-5-1-2	7554	18 27 54 14 5 7 30 41 63 49 17 69 26 52 79 81 55 13 20 33 85 21 90 4 10 96 1 64 37 98 92 35 16 93 48 58 2 25 68 83 31 74 22 82 40 45 9 50 91 29 3 89 46 78 57 71 36 34 77 65 12 24 95 15 39 97 38 88 72 94 84 67 59 8 80 60 44 61 76 19 99 51 42 66 70 62 73 75 86 87 6 100 56 53 32 28 11 23 43 47
100-5-1-3	7797	34 42 8 57 58 37 10 12 95 11 28 71 17 40 81 64 51 15 49 59 43 25 33 44 45 52 4 56 100 36 84 77 29 23 22 16 53 85 47 9 55 2 93 31 97 41 48 3 27 72 46 83 35 1 61 30 63 65 68 24 5 80 98 62 86 14 21 74 99 26 66 69 90 38 73 91 92 88 32 94 13 20 54 67 7 6 50 78 19 79 39 89 96 82 18 60 70 75 87 76
100-5-2-1	7777	72 86 63 46 83 76 12 84 31 78 54 18 67 2 22 43 59 1 9 27 30 15 5 50 100 99 81 82 24 96 20 23 25 37 29 32 47 64 51 33 91 74 35 8 85 34 14 77 79 41 17 38 45 4 52 21 69 55 62 75 44 95 10 65 66 94 92 13 71 58 98 48 90 7 68 61 28 40 87 88 3 80 11 89 56 60 16 39 93 97 36 49 19 57 6 73 26 70 53 42
100-5-2-2	8019	52 15 58 76 25 74 31 75 97 84 8 27 46 68 94 44 92 24 21 10 33 54 85 60 11 70 96 40 42 1 91 18 100 34 2 48 80 3 57 41 16 50 26 22 86 55 32 99 30 43 71 39 49 59 19 72 28 98 93 14 47 69 81 77 29 20 23 38 35 90 45 61 83 87 65 5 51 66 56 37 95 62 64 13 4 63 12 36 9 89 53 88 78 73 79 7 82 67 6 17
100-5-2-3	7607	47 84 44 30 38 72 85 20 42 69 37 31 12 94 39 70 52 88 83 79 92 75 1 25 24 28 26 17 77 36 78 90 50 27 76 45 15 59 60 51 71 10 13 57 54 35 32 55 100 81 14 66 67 86 80 11 49 93 65 33 91 41 97 58 29 68 82 96 89 23 40 6 43 16 95 46 9 73 4 5 99 19 3 64 98 53 56 48 61 87 63 74 62 7 8 34 21 18 2 22
150-6-1-1	11839	148 103 47 48 20 143 86 142 5 59 130 29 128 37 134 57 138 144 46 132 11 117 81 49 107 56 108 67 141 44 150 90 78 12 136 106 30 88 104 8 87 84 1 58 116 139 140 91 73 70 92 114 26 99 149 42 10 50 85 131 34 118 100 97 60 64 53 52 27 39 96 115 123 105 110 40 135 102 43 71 21 55 109 94 129 89 28 120 95 121 17 36 2 126 83 6 93 146 16 68 15 51 145 62 7 14 33 147 4 13 69 80 125 72 66 124 61 31 18 24 111 112 119 23 137 35 122 101 25 22 74 75 45 19 65 76 82 38 63 54 113 98 133 127 32 3 79 41 77 9
150-6-1-2	12056	110 37 150 47 138 91 136 121 54 130 38 42 7 126 75 141 17 23 145 32 147 86 55 88 2 139 95 104 66 29 142 120 116 149 19 94 63 107 101 119 85 148 28 44 39 137 16 123 53 31 77 58 8 56 113 27 125 36 48 65 81 122 68 83 87 21 41 129 131 92 13 71 51 143 33 49 11 50 72 82 146 60 100 102 1 103 96 109 4 24 117 128 115 45 26 52 127 79 98 9 111 64 112 57 99 15 93 124 10 5 108 97 30 25 59 67 22 3 84 105 20 18 35 70 90 80 76 144 118 89 140 106 6 133 46 114 135 132 40 14 34 61 134 74 12 43 73 78 69 62
150-6-1-3	12218	71 3 145 84 73 94 112 8 24 63 50 117 81 140 22 127 55 10 121 109 104 19 138 1 69 133 99 11 25 123 30 137 125 31 107 20 88 21 82 96 150 106 119 97 56 89 64 105 66 38 37 29 18 103 42 141 114 28 79 108 60 27 131 53 58 5 15 75 126 78 67 70 110 48 80 54 23 74 43 87 136 146 9 147 76 143 101 95 4 122 111 2 120 45 134 86 92 68 113 130 34 39 13 40 139 26 116 57 7 12 115 35 41 142 14 32 72 135 49 44 47 16 65 6 91 128 118 77 83 144 129 98 51 52 132 17 100 149 36 90 102 46 124 85 33 148 62 61 59 93
150-6-2-1	12308	4 24 141 94 102 130 28 150 131 46 77 121 5 19 51 32 78 16 149 58 57 13 109 134 107 68 23 29 140 44 60 100 54 39 18 138 71 85 91 120 3 136 59 10 81 97 45 129 118 110 20 66 76 74 101 88 148 111 114 119 98 112 84 21 133 93 147 89 95 31 7 79 8 26 145 139 30 126 142 117 146 125 92 37 72 65 127 64 40 99 96 55 116 86 61 137 123 124 25 73 82 11 34 43 90 108 143 106 14 50 38 9 48 33 70 41 132 49 87 105 1 135 144 83 6 27 12 128 67 80 52 36 104 42 47 15 122 17 69 53 113 56 62 2 35 63 75 115 103 22
150-6-2-2	12310	77 149 18 53 81 95 39 123 25 108 3 17 140 103 107 114 38 111 29 102 130 145 63 96 141 14 110 31 34 116 89 104 23 120 7 42 33 93 60 85 126 92 118 119 127 124 2 27 88 64 47 139 8 41 125 67 105 82 144 11 87 19 100 15 4 150 135 71 117 146 16 106 134 148 76 54 52 70 13 109 44 45 83 66 32 98 147 80 43 137 99 84 55 69 94 21 28 74 97 112 35 22 49 86 143 65 24 12 128 115 46 91 9 20 1 50 129 5 61 90 138 142 121 51 37 73 57 10 48 68 133 122 136 30 79 58 40 101 131 26 56 36 78 132 75 59 113 72 62 6
150-6-2-3	12048	143 148 9 150 72 12 87 94 34 2 138 24 91 147 69 101 110 25 125 4 32 33 127 137 36 108 58 120 67 16 11 42 86 61 70 53 119 115 21 57 6 66 133 8 144 31 116 41 65 88 96 62 44 129 95 102 112 29 85 126 111 81 93 131 132 46 51 106 113 82 60 30 48 84 80 23 103 37 97 26 79 3 76 1 99 92 10 122 135 13 71 52 64 39 35 124 89 98 140 142 68 59 74 18 27 14 19 49 7 134 55 50 117 17 141 118 130 114 100 56 123 105 54 15 40 28 104 145 107 139 22 128 90 121 43 20 77 136 5 146 47 63 149 45 109 75 78 38 73 83