Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 13, No. 2, 2021 Article ID IJIM-1124, 18 pages
DOR: http://dorl.net/dor/20.1001.1.20085621.2021.13.2.6.1
Research Article
Science and Research Branch (IAU)

Due Date Assignment and JIT Scheduling Problem in Blocking Hybrid Flow Shop Robotic Cells with Multiple Robots and Batch Delivery Cost

J. Rezaeian *†, N. Derakhshan ‡, I. Mahdavi §, R. Alizadeh Foroutan ¶

Received Date: 2017-09-20 Revised Date: 2018-07-11 Accepted Date: 2021-01-25

Abstract

In this paper, just-in-time scheduling problem with batch delivery and due date assignment for hybrid flow shop robotic cells is considered. A mixed integer linear programming (MILP) model is presented to determine the sequence of jobs and robot moves. Two meta-heuristic algorithms including Artificial Immune System (AIS) and Tabu Search (TS) are proposed. The results show that AIS performs well for this problem.

Keywords : JIT scheduling; Robotic cells; Multiple robots; Batch delivery; Due date assignment.

1 Introduction

One of the most important scheduling problems that is commonly encountered in manufacturing environments is hybrid flow shop scheduling (HFS) problem. The HFS consists of two or more stages that at least one of them has two or more parallel machines which can be identical, uniform or unrelated. During the last decade, researches in this field have been focusing on more realistic problems including machine eligibility, blocking constraints, transportation constraints, etc. When machine eligibility constraints are considered, machines are not eligible to process all the jobs at stages. It means that not all jobs can be processed on all machines. Applying blocking constraints is a well-known characteristic which is widely considered in scheduling problems. In real life situation, there is no buffer space between two successive machines for jobs to wait for next operation; this causes the job that has completed its processing, cannot move from the current machine to the other which is occupied. The most effective factors in increasing the productivity of manufacturing systems are material handling time and cost. Since these factors make bottleneck, efficient material handling has become very important. Today, many industries use computer-controlled material handling

*Corresponding author. j_rezaeian@ustmb.ac.ir, Tel:+98(911)1571257.
†Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.
‡Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.
§Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.
¶Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.

systems such as robots for this purpose. Industrial robots play an important role in advanced manufacturing systems [19]. In these manufacturing systems, job loading, unloading and transportation are done by robots. Indeed, in advanced manufacturing systems, cost reduction is the main purpose of manufacturers. With respect to cost reduction, the most important scheduling problem that has received considerable attention in last decades is just in time (JIT) scheduling. In fact, a vast variety of practical problems are related to just in time (JIT) scheduling environment. Because of practical importance and industrial application of JIT scheduling, investigations on scheduling problems include both earliness and tardiness penalty. In most scheduling models, a job that is completed after its due date, incurs a tardiness penalty due to customer dissatisfaction; a possible contractual cost for late delivery and potential loss of reputation. On the other hand, the completion of jobs before their due date could result in additional storage or insurance costs, or even product deterioration [20].

One of the principal features of JIT scheduling relies on the way the due dates are considered: they can be given parameters or decision variables [24]. While traditional scheduling models considered due dates as given by exogenous decisions, in an integrated system, they are determined by taking into account the systems ability to meet the quoted delivery dates. In order to avoid tardiness penalties, including the possibility of losing customers, companies are under increasing pressure to quote attainable delivery dates. At the same time, promising delivery dates too far into the future may not be acceptable to the customers or may force a company to offer price discounts in order to retain the business. Thus, there is an important trade-off between assigning relatively short due- dates to customer orders and avoiding tardiness penalties. This is the reason that increasingly large number of recent studies have regarded due date assignment as a part of scheduling process, and shows how the ability to control due dates can be a major factor in improving system performance [20]. The initial researches in the field of due date assignment in scheduling focused on the constrained version where the scheduler must decide on a common due date for all jobs. This method is usually referred to as the CON method; while dealing with the unrestricted case, each job can have a different due date which is referred to as the DIF method [22].

Considering the variety of production and flexibility of manufacturing systems, the due date assignment of product batches and consequently the ability to deliver them just in time is one of the key factors in todays competitive world such that trying to adjust its production scheduling with customers order is inevitable for every company to survive. Therefore, JIT scheduling has emerged as a response to the necessity of fulfilling each customers order at his most desired time. On the other hand, the accurate handling flow is one of the fundamental characteristics of productivity in advanced serial manufacturing systems. Hence, taking these two concepts into account together which means keep being customer-oriented in order to acquire their satisfaction and meet companys potential share of market and simultaneously managing material handling time and cost in a production plant, makes robots employment very reasonable. With these motivations, utilizing robots aroused and constantly has become more and more applied in all types of modern production systems and their corresponding sequencing and scheduling models. Blocking hybrid flow shop robotic cells scheduling (BHFS-RCS) is a notable development on the well-known classic hybrid flow shop scheduling problem. Although BHFS-RCS have been increasingly studied during last decade, but there has not been any research yet where the strongly connected due date assignment and JIT scheduling are considered mutually together as an objective along with batch delivery costs. Most researchers treated delivery costs as either negligible or irrelevant, however, delivery costs are a significant factor, and hence production costs depend not only on when jobs are processed but also when finished jobs are delivered [20].

This research addresses blocking hybrid flow shop scheduling problem with transport resources as robots in JIT environment. In fact, in this paper we study blocking hybrid flow shop robotic cell scheduling problem incorporating additional constraints such as unrelated parallel machines and machine eligibility constraints with respect

to JIT scheduling, due date assignment by DIF method and batch delivery. We considered this problem with unloading, transferring and loading jobs which are performed by robots. According to our survey on the literature which is described in detail in the next section, the BHFS-RCS problem as explained above has not been studied in the literature before and there is a gap here.

The rest of the paper is organized as follows: the related literature is reviewed in the next section. Section 3, gives the definition of the problem and details of the proposed MILP model. In section 4, AIS and TS algorithms are proposed. Section 5 presents the computational experiments and analysis and finally conclusion of this research and future extensions are given in Section 6.

2 Literature review

Most of the literature of robotic cells scheduling deals with cyclic scheduling in which the objective is to maximize the throughput rate, or equivalently, minimize the cycle time. For this reason, minimizing makespan has been considered as maximizing throughput in the literature.

Bilge and Ulusoy [3] studied the simultaneous scheduling of automatic guided vehicles (AGVs) in flexible manufacturing system. They developed an iterative procedure in which the objective is makespan minimization. Hurink and Knust [14] considered flow shop scheduling with transportation times and a single robot. They assumed an unlimited buffer space between the machines to determine a feasible schedule while minimizing the makespan. Furthermore, Hurink and Knust [13] proposed a Tabu search algorithm for the job shop scheduling problem with a single transportation resource. Soukhal and Martineau [23] studied the flow shop robotic cell scheduling problem with multiple part-types and a single transportation robot. They assumed that there is no intermediate buffer between machines and proposed a genetic algorithm to solve the problem. Carlier et al. [4] considered an approximate decomposition algorithm for the flow shop automated cells scheduling problem with a single transportation robot and a blocking constraint. Their presented approach divides the problem into two separate scheduling problems which are solved sequentially. To solve each of these two problems, they proposed an exact branch-and-bound algorithm and a two-phase genetic algorithm. Kharbeche et al. [15] studied the flow shop robotic cell scheduling problem with single robot. They proposed an exact branch-and-bound algorithm to solve the problem and also a genetic algorithm to deal with large-scale problems. Geismer et al. [11] considered the bufferless robotic cells flow shop with parallel machines and constant travel time. Geismer et al. [12] also studied the same problem with multiple robots.

Elmi and Topaloglu [6] considered blocking hybrid flow shop robotic cells scheduling problem in which the objective is to minimize the makespan. They proposed simulated annealing (SA) algorithm to solve the problem. Elmi and Topaloglu [9] also studied the robotic scheduling problem considering multiple part-types in a blocking hybrid flow shop cells environment including different speed parallel machines at each stage, machine eligibility constraints and a single robot to convey parts between stages. They developed a SA solution approach to solve the proposed MILP model. Batur et al. [2] focused on a hybrid flexible flow shop robotic cells scheduling problem considering multiple part-type production and tried to determine the best cycle time and the robot move sequence which transports parts between machines. They presented a SA based meta-heuristic and solved the problem using two different neighborhood structures. Arviv et al. [1] studied two-robot job transfer flow-shop scheduling problem with the aim of minimizing makespan. In order to achieve that, they constructed a new collaborative reinforcement learning algorithm using dual Q-learning functions. Shabtay and Arviv [21] considered a non-cyclic three-machine robotic flow-shop scheduling problem. A single robot is assumed for transferring jobs between machines. The objective is to minimize the makespan. They solved the problem by decomposing it into a set of sub-problems and provide an optimal schedule for each of them.

Elmi and Topaloglu [8] investigated multi-degree cyclic flow shop robotic cell scheduling problem considering multiple gripper robots and developed an ant colony optimization (ACO)

meta-heuristic to determine the optimal sequence of robots' moves along with the optimal degree of the cyclic schedule and the robot assignments for the transportation operations. Che et al. [5] developed a bicriteria polynomial algorithm to find the Pareto front for a stable robotic flow shop scheduling with interval times and a fixed robot route. Their objectives were cycle time minimization and stability radius maximization at the same time. Zabihzadeh and Rezaeian [26] studied flexible flow shop scheduling problem containing release time and robotic transportation. They presented a MILP model to minimize the maximum completion time of all parts and proposed two meta-heuristics including SA and ACO to solve the problem. Lei et al. [16] proposed a hybrid algorithm based on a hybrid quantum evolutionary algorithm to solve the cyclic scheduling problem with single robot and flexible processing time. Another study by Elmi and Topaloglu [7] concentrated on multi-degree cyclic flow shop robotic cell scheduling problem in which multiple robots are supposed to handle the transportation operation of parts between machines. A new MILP model was proposed by them to maximize the throughput rate.

To the best of the authors knowledge, JIT scheduling along with due date assignment and batch delivery have not been considered together in hybrid flow shop robotic cells systems in the literature and therefore this research is quite innovative in the area trying to fill this gap. This paper considers earliness, tardiness, due date assignment and batch delivery costs in these systems as objective function and unrelated parallel machines, multiple part types and machine eligibility as constraints.

3 Problem statement and formulation

In this section, first we present definitions and notations of the proposed problem; next a mixed integer linear programming model is described. The manufacturing environment of the BHFS-RCS is considered as an extension of the flow shop in which the handling resource is multiple robots. In a robotic flow shop cell with parallel machines, there are J jobs that need to be processed at all stages in the same order, starting at stage 1 until finishing in stage S. Each stage s consists of a given number of unrelated parallel machines M_s ($M_s \geq 1$). Each job j can be processed at the sth stage by any machine from the eligible machine set which is denoted by $E_{j,s}$. The transfer of a job j from one machine to another is performed by robots. In each stage, there is no buffer space between successive machines. It means that after processing a job j on one machine in stage s, the machine remains blocked until a robot picks up job j and transfers it to another machine in the next stage.

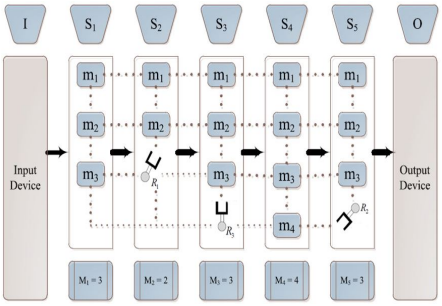

Figure 1: An example of a BHFS-RCS with parallel machines considered in this research.

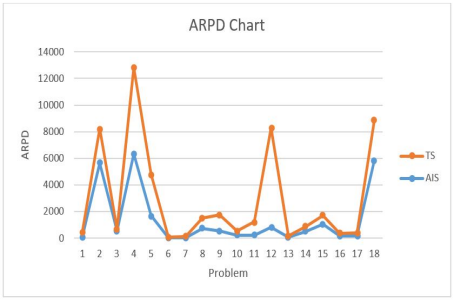

Figure 2: ARPD comparison of proposed algorithms.

Fig. 1 presents the considered BHFS-RCS problem with unrelated parallel machines at each stage. In this figure, trapezoids show the stages, rectangles present the machines placed at each stage, the disjunctive lines show transporting robots' movements between machines to load or unload them, the conjunctive arcs illustrate the processing sequence of parts throughout the stages and finally three robots (noted as R_1, R_2 and R_3) on their paths between machines are depicted as manipulators.

Table 1: Solution encoding in artificial immune system.

Operation	2	4	7	9
Job	1	1	1	1
Stage	1	2	3	4
Batch	2	2	2	2

Table 2: Considered parameters in [6].

Parameters	**Values**
The number of jobs	10
The number of stages	4,6,8
The number of machines in each stage	U(6,1)
The number of robots	2,4,6,8,10
The processing times	U(10,100)
The processing speeds	U(1,3)

Table 3: Considered parameters in computational experiments.

Parameters	**Values**
The number of jobs	10,20
The number of stages	4,6,8
The number of machines in each stage*	U(1,4)
The number of robots	2,4,6,8,10
The number of batches	3
The processing times	U(10,100)
The processing speeds	U(1,3)
The acceptable lead time of jobs	$\boldsymbol{round}\left(\boldsymbol{U}(0,1) * \sum_{i=1}^{stage} \boldsymbol{P}_{ij}\right) + \sum_{i=1}^{stage} \boldsymbol{P}_{ij} \forall j$
cost per batch delivery	U(1,20)
earliness penalty weight per job	U(1,20)
tardiness penalty weight per job	U(1,20)
due date penalty weight per job	U(1,20)

* All of these machines are not eligible to process all jobs; hence the number of eligible machines is generated randomly.

For each job there is an acceptable lead time, that is reasonable for its customer who doesn't want to receive his order earlier or later than due date. Hence, if due date won't be greater than the acceptable lead time, there will be no penalty.

After finishing job processing, the finished jobs are to be delivered to customer in batches. A delivery batch is defined as a set of jobs with common due dates. There is no capacity limitation on a batch delivery. The delivery time of all jobs in a batch is equal to the maximum departure time of them from last stage. If a job is delivered

Table 4: Problems categorized into three intervals.

	$\boldsymbol{Size\ of\ solution = job * (stage - 1)}$		
Problem size	30-60	70-100	110-140
small	✓		
average		✓	
large			✓

Table 5: Considered levels of AIS parameters

2*AIS parameters	$\boldsymbol{Size\ of\ solution = job * (stage - 1)}$		
	30-60	70-100	110-140
3**maxgen*	100	100	100
	120	120	120
	150	150	150
3**Pop size*	50	50	50
	60	60	60
	70	70	70
3*n_c	7	7	7
	8	8	8
	9	9	9

Table 6: Considered levels of TS parameters.

21.6cmTS parameters	$\boldsymbol{Size\ of\ solution = job * (stage - 1)}$		
	30-60	70-100	110-140
3**iteration*	100	100	110
	150	110	120
	200	120	130
3**TL*	*Round (0.3* * *nAction)*	*Round (0.6 * nAction)*	*Round (0.75 * nAction)*
	*Round (0.4 * nAction)*	*Round (0.7 * nAction)*	*Round (0.85 * nAction)*
	*Round (0.5 * nAction)*	*Round (0.8 * nAction)*	*Round (0.9 * nAction)*

* Number of allowable moves in neighborhood.

Table 7: Considered levels of AIS parameters

2*AIS parameters	$\boldsymbol{Size\ of\ solution = job * (stage - 1)}$		
	30-60	70-100	110-140
maxgen	100	100	100
Pop size	50	50	50
n_c	7	7	7

before the due date, it has to be held until its due date and incurs earliness penalty. On the other hand, if a job is delivered after the due date, it causes tardiness penalty. The objective is to minimize sum of earliness penalty, tardiness penalty, due date assignment and batch delivery costs.

Table 8: Considered levels of TS parameters

21.6cm**TS parameters**	$\boldsymbol{Size\ of\ solution = job * (stage - 1)}$		
	30-60	70-100	110-140
iteration	100	110	110
TL	*Round (0.3 * nAction)*	*Round (0.8 * nAction)*	*Round (0.85 * nAction)*

Table 9: The effectiveness of the proposed algorithms in comparison with SA algorithm of Elmi's paper [6].

2***jobs**	2***stages**	2***algorithm**	**Number of robots (NR)**				
			2	4	6	8	10
9*10	3*4	AIS	757.60	426.96	295.73	279.49	282.90
		TS	724.16	456.39	407.60	305.86	266.53
		SA	804.54	584.63	521.48	523.64	519.31
	3*6	AIS	820.20	682.89	530.16	480.96	426.83
		TS	1171.93	686.89	584.19	469.56	527.19
		SA	874.48	798.96	737.84	646.74	643.71
	3*8	AIS	1191.33	993.83	793.70	657.03	620.36
		TS	1384.33	1018.43	843.29	783.53	623.50
		SA	1201.74	1033.58	925.56	897.82	862.75

Table 10: The comparison of GAP between proposed algorithms and Elmi's SA algorithm [6].

multirow2***jobs**	2***stages**	2***algorithm**	**Number of robots (NR)**				
			2	4	6	8	10
6*10	3*4	AIS	5.83	26.96	43.29	46.62	45.52
		TS	9.99	21.93	21.83	41.58	48.67
	2*6	AIS	6.20	14.52	28.14	25.63	33.69
		TS	0	14.02	20.82	27.39	18.10
	2*8	AIS	0.86	3.84	14.24	26.81	23.92
		TS	0	1.46	8.88	12.72	27.73

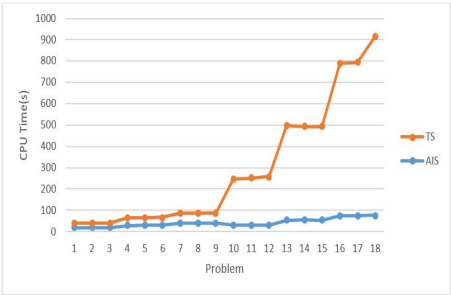

Figure 3: CPU time comparison of proposed algorithms.

3.1 Notation

Indices:

J= number of jobs, $j \in \{1, 2, \ldots, J\}$.

S= number of stages, $s \in \{1, 2, \ldots, S\}$.

M_s= number of machines at stage s, such that there is one machine in input and output stages, $m \in \{1, 2, \ldots, M_s\}$.

W= number of operations that are performed by robots, $f \in \{1, 2, \ldots, W\}$.

B= number of batch, $b \in \{1, 2, \ldots, B\}$.

Table 11: Randomly generated problems using data from Table 2.

Problem	Number of jobs	Number of stages	Number of robots	Number of batchs
1	10	4	2	3
2	10	4	4	3
3	10	4	6	3
4	10	6	2	3
5	10	6	4	3
6	10	6	6	3
7	10	8	2	3
8	10	8	4	3
9	10	8	6	3
10	20	4	2	3
11	20	4	4	3
12	20	4	6	3
13	20	6	2	3
14	20	6	4	3
15	20	6	6	3
16	20	8	2	3
17	20	8	4	3
18	20	8	6	3

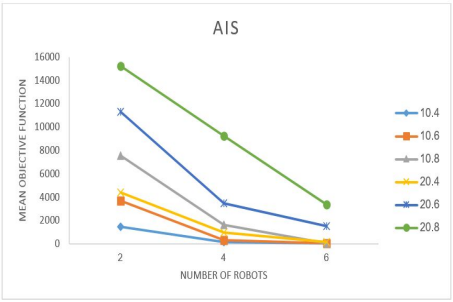

Figure 4: Average objective reduction using different numbers of robots for each problem in AIS algorithm.

Parameters:

$P_{j,s}$= processing time of the jth part at stage s, where the processing times of all jobs at input and output stages are equal to zero.
$R_{s,m}$= processing speed of the mth machine at stage s, where the speed of the machines at input and output stages are zero.
$E_{j,s}$= number of machines that are eligible to process the jth job at stage s, $m \in \{1, 2, \ldots, E_{j,s}\}$.

A_j= acceptable lead time for job j.
α_j= earliness penalty weight for job j.
β_j= tardiness penalty weight for job j.
γ_j= due date penalty weight for job j.
θ= cost per batch delivery.
SP= the time required for the robot to travel each distance.
BM= a very large number.

Decision Variables:

$X_{j,j',s}$=1 if the j'th part is processed after the jth job at stage s, 0 otherwise.
$Y_{j,s,m}$=1 if job j is processed on machine m at stage s, 0 otherwise.
$Z_{j,s,f}$=1 if job j at stage s is the fth operation that would be unloaded by any robot to be transported to the next stage.
$RS_{r,f}$=1 if the fth operation is unloaded by rth robot, 0 otherwise.

Table 12: ARPD in terms of objective value for proposed algorithms.

Problem	$Best_{sol}$	AIS_{sol}	Time Spent$_{sol}$	TS_{sol}	Time Spent$_{TS}$
1	646.33	126.34	17.44	368.55	**22.21**
2	3	5646.66	17.8	2580	**22.71**
3	12	573.33	17.96	105	**22.91**
4	57	6348.53	28.72	6480.52	**37.24**
5	18	1691.09	29.39	3111.47	**37.2**7
6	24	47.5	29.6	50	**37.2**
7	4974.83	51.5	40.48	98.56	**46.29**
8	194.33	747.6	40.9	784.51	**45.95**
9	12	570.83	40.99	1188.88	**46.54**
10	1284.33	242.92	30.48	298.55	**217.59**
11	287	240.77	31.21	1005.29	**218.99**
12	18	840.54	31.87	7436.66	**226.68**
13	6241	80.97	53.04	107.82	**443.83**
14	561.66	518.95	54.19	367.94	**438.43**
15	129.33	1084.5	53.83	613.97	**437.18**
16	5581.66	172.54	72.97	202.92	**717.58**
17	3161	191.9	75.32	233.7	**718.31**
18	57	5820.51	76.85	3041.62	**839.23**

Table 13: The average objective value for proposed algorithms.

2*Problem		AIS			TS		
		Number of robots					
10	4	1482.93	172.4	80.8	3028.53	80.4	24.6
10	6	3675.66	322.39	35.4	3750.9	578	36
10	8	7537.09	1647.16	80.5	9878.46	1718.89	154.66
20	4	4404.33	978.03	169.29	5118.76	3172.2	1356.6
20	6	11294.59	3476.43	1531.93	12970.6	2628.36	923.39
20	8	15212.96	9227.16	3374.69	16908.2	1048.6	1449.53

$YY_{j,b}$=1 if job j belongs to batch b, 0 otherwise.
ZZ_b=1 if there is at least one job in batch b. 0 otherwise.
$D_{j,s}$= the departure time of the jth job from stage s.
dd_j= due date of job j.
DT_j= delivery time of job j.
T_j= tardiness of job j, which is equal to $\max\{0, DT_j - dd_j\}$.
E_j= earliness of job j, which is equal to $\max\{0, dd_j - DT_j\}$.
G_j= delay in delivery of job j which is equal to $\max\{0, dd_j - A_j\}$.

Objective function and constraints:

Minimize:

$$\sum_{j=1}^{J}\alpha_j T_j+\sum_{j=1}^{J}\beta_j E_j+\sum_{j=1}^{J}\gamma_j G_j+\sum_{b=1}^{B}\theta Z Z_b$$

s.t.

$$D_{j,s}+BM\times\Big(2-(Y_{j,s,m}+Y_{j,s-1,m'})\Big)\geq D_{j,s-1}+\Bigg(SP\times(|s-(s-1)|+|m-m'|)\Bigg)+(P_{j,s}\times R_{s,m}),$$
$$i\in\{1,2,\ldots,J\};\ s\in\{2,\ldots,S-1\});\quad m\in\{1,2,...,E_{s,j}\};\ m'\in\{1,2,...,E_{j,s-1}\}\quad(3.1)$$

$$\left(\sum_{m=1}^{E_{j,s}}Y_{j,s,m}\right)=1\qquad j\in\{1,2,\ldots,.J\};\ s\in\{1,2,\ldots,S\}\qquad(3.2)$$

$$\left(\sum_{m=E_{js+1}}^{M_s}Y_{j,s,m}\right)=0\qquad j\in\{1,2,\ldots,J\};\ s\in\{1,2,\ldots,S\}\qquad(3.3)$$

$$D_{j,s-1}+BM\times(4-Y_{j,s,m}-Y_{j',s,m}-Y_{j',s-1,m'}-X_{j,j',s})\geq D_{j,s}-\Big(SP\times(1-|m-m'|)\Big)$$
$$j,j'\in\{1,2,...,J\},j\neq j';\ s\in\{2,...,S-1\};\quad m\in\{1,2,...,E_{j,s}\};\ m'\in\{1,2,...,E_{j',s-1}\}\quad(3.4)$$

$$D_{j,s-1}+BM\times(3-Y_{j,s,m}-Y_{j',s,m}-Y_{j,s-1,m'}+X_{j,j',s})\geq D_{j',s}-\Big(SP\times(1+|m-m'|)\Big)$$
$$j,j'\in\{1,2,...,J\},j\neq j';\ s\in\{2,...,S-1\};\quad m\in\{1,2,...,E_{j,s}\};\ m'\in\{1,2,...,E_{j,s-1}\}\quad(3.5)$$

$$\left(\sum_{f=1}^{W}Z_{j,s,f}\right)=1\qquad j\in\{1,2,\ldots,J\};\ s\in\{1,2,\ldots,S-1\}\qquad(3.6)$$

$$\left(\sum_{j=1}^{J}\sum_{s=1}^{S-1}Z_{j,s,f}\right)=1,\quad f\in\{1,2,\ldots,W\}\quad(3.7)$$

$$\left(\sum_{f=1}^{f'}Z_{j,s,f}\right)-BM\times(1-Z_{j,s-1,f'})\leq 0\qquad j\in\{1,2,\ldots,J\};\ s\in\{2,\ldots,S-1\};\quad f'\in\{1,2,\ldots,W\}\qquad(3.8)$$

$$\left(\sum_{f=1}^{f'}Z_{j,s-1,f}\right)-BM\times(3-Y_{j',s,m}-Y_{j,s,m}-Z_{j',s,f'}+X_{j,j',s})\leq 0$$
$$j,j'\in\{1,2,...,J\},j\neq j';\ s\in\{2,...,S-1\};\quad m\in\{1,2,\ldots,E_{j,s}\};\ f'\in\{1,2,\ldots,W\}\qquad(3.9)$$

$$\left(\sum_{f=1}^{f'}Z_{j',s,-1,f}\right)-BM\times(4-Y_{j',s,m}-Y_{j,s,m}-Z_{j,s,f'}-X_{j,j',s})\leq 0$$
$$j,j'\in\{1,2,...,J\},j\neq j';\ s\in\{2,...,S-1\};\quad m\in\{1,2,\ldots,E_{j',s}\};\ f'\in\{1,2,\ldots,W\}\qquad(3.10)$$

$$\sum_{r=1}^{R}RS_{rf}=1\qquad f\in\{1,2,\ldots,W\}\qquad(3.11)$$

$$D_{j,s}+\Big(BM\times(7-RS_{r,f}-RS_{r,f'}-Z_{j,s,f}-Z_{j',s',f'}-Y_{j,s,m}-Y_{j',s',m'_1}-Y_{j',s'+1,m'_2})\Big)\geq D_{j',s'}+SP\times\Big((|s'-(s'+1)|+|m'_1-m'_2|)+(|(s'+1)-s|+|m'_2-m|)\Big)$$
$$j,j'\in\{1,2,\ldots,J\}\ \ j\neq j';\quad s,s'\in\{1,2,...,S-1\};\ m\in\{1,2,...,E_{j,s}\};\quad m'_1\in\{1,2,...,E_{j',s'}\};\quad m'_2\in\{1,2,...,E_{j',s'+1}\};\quad f,f'\in\{2,;W\},f<f';\ r\in\{2,\ldots,R\}\qquad(3.12)$$

$$\sum_{b=1}^{B}YY_{j,b}=1\qquad j\in\{1,2,\ldots,J\}\qquad(3.13)$$

$$DT_j \geq D_{j',S-1} + (SP \times (|S-(S-1)|+ |1-m|)) - BM \times (2 - YY_{j,b} - YY_{j',b}) \quad j, j' \in \{1,2,\ldots,J\};\ b \in \{1,2,\ldots,B\}; \quad m \in \{1,2,\ldots,E_{j',S-1}\} \tag{3.14}$$

$$dd_j \geq dd_{j'} - BM \times (2 - YY_{j,b} - YY_{j',b}) \quad j, j' \in \{1,2,\ldots,J\};\ b \in \{1,2,\ldots,B\} \tag{3.15}$$

$$dd_j \leq dd_{j'} + BM \times (2 - YY_{j,b} - YY_{j',b}) \quad j, j' \in \{1,2,\ldots,J\};\ b \in \{1,2,\ldots,B\} \tag{3.16}$$

$$DT_j + E_j - T_j = dd_j \qquad j \in \{1,2,\ldots,J\} \tag{3.17}$$

$$ZZ_b \geq YY_{j,b} \quad j \in \{1,2,\ldots,J\};\ b \in \{1,2,\ldots,B\} \tag{3.18}$$

$$ZZ_{b-1} \geq ZZ_b \qquad b \in \{2,\ldots,B\} \tag{3.19}$$

$$G_j \geq dd_j - A_j \qquad j \in \{1,2,\ldots,J\} \tag{3.20}$$

$$G_j, T_j, E_j \geq 0 \qquad j \in \{1,2,\ldots,J\} \tag{3.21}$$

Constraint 3.1 ensures that if a job is processed at a stage it must also be processed at the previous stage. Constraints 3.2 and 3.3 state that, only one of the eligible machines at each stage is able to process a job. Constraints 3.4 and 3.5 determine the sequence of job processing operations on the same machine at each stage. These constraints ensure that a machine isnt able to process two jobs at the same time. Constraints 3.6 and 3.7 indicate that each robot is able to transfer only one job from a stage to the next one. Constraint 3.8 ensures that a job can be transferred to current stage only if it was completed in previous stage. Constraints 3.9 and 3.10 state that, the sequence of job processing operations determines the sequence of robots move. These constraints guarantee that each job is able to be loaded on a machine, only if a previous job was unloaded from it. Constraint 3.11 ensures that only one robot is assigned for each unloading operation. Constraint 3.12 states that the two unloading operations are not able to be handled at the same time. Constraint 3.13 prevents assigning more than one batch to each job. Constraint 3.14 states that the delivery time of each job in a batch is equal to maximum departure time of all jobs in the same batch from last stage. Constraints 3.15 and 3.16 express that all jobs in the same batch have equal due dates. Constraint 3.17 ensures that the delivery time of each job should be determined according to the earliness, tardiness and due date of that job. Constraints 3.18 and 3.19 determine the number of batches. Constraint 3.20 indicates that the delay of each job is greater than minimum deviation of jobs due date from its acceptable lead time. Constraint 3.21 ensures that the earliness, tardiness and delay of each job is greater than or equal to zero.

The large number of decision variables and constraints of the proposed model cause the exact methods not to be able to solve it for small-size problems. Hence artificial immune system and Tabu search algorithms are developed in the next sections to solve this issue.

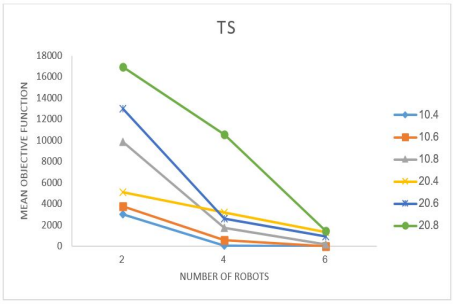

Figure 5: Average objective reduction using different numbers of robots for each problem in TS algorithm.

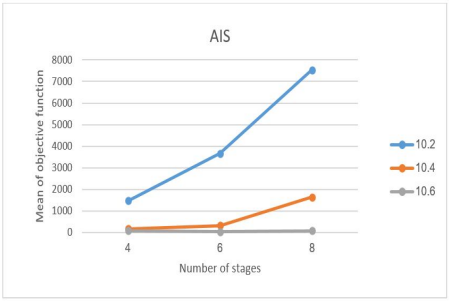

Figure 6: Average objective rise using different numbers of stages for each problem in AIS algorithm

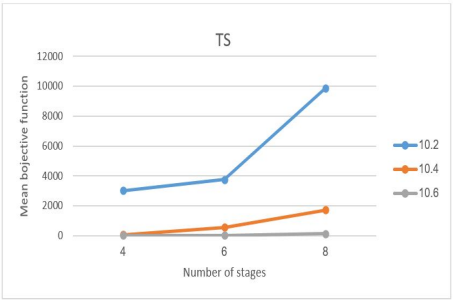

Figure 7: Average objective rise using different numbers of stages for each problem in TS algorithm.

4 The proposed algorithms

4.1 Artificial immune system

The operative mechanisms of immune system are very important; hence the proposed artificial immune system in this paper is based on clonal selection and affinity maturation.

The main task of immune system is to identify and destroy viruses, bacteria and other foreign substances which are denoted as pathogens. These pathogenic substances are recognized by structure molecules on their surfaces which are known as antigens. After identifying antigens, the immune system develops defensive mechanisms against them. Hence the most common types of immune cells which are called lymphocytes (B cells and T cells) interact to initiate antibody production. These antibodies immobilize antigens and prevent them from causing infection.

In other words, pathogens will be swallowed by phagocytes and their antigen is represented on these cells' surface. The lymphocytes have receptor molecules on their surface which is able to recognize antigens. When antigen and receptor molecules have complementary shape, they can bind together; this binding ensures recognition of antigen [10]. The evaluation of the receptor matching is called affinity. The procedure from the intrusion to the recognition of pathogens is called the primary immune response.

T cells that match the antigen are proliferated and differentiated to kill the pathogenic substances and so do B cells, and then secrete antibodies into body. After that, if a similar antigenic substance enters the human body again, the memory B cells will be quickly activated at a high stimulation level so that the human body can effectively respond to the recognized or similar pathogen. This process is called the secondary immune response [25]. Therefore, the entire process from the intrusion to the elimination of the pathogens is called the immune response.

According to the clone selection procedure, this algorithm starts searching with generating population of antibodies. These antibodies indicate the feasible solutions. The objective of these solutions is called antigens. After generating antibodies, each antibody is evaluated according to its affinity measure. Then all antibodies are sorted according to their affinity measures. After sorting, defined number of high affinity antibodies are selected to proliferate according to their affinity measure. The clone number of an antibody is proportional to its affinity measure. Then the clones mutate according to their origin's affinity measures, the clones of the fitter origin will suffer less mutation [25]. Finally, after determining the affinity measures of the mutants, the defined number of high affinity mutants would be replaced with low affinity population of antibodies. The whole processes are repeated until the stop criteria are satisfied. The proposed algorithm is summarized below: Initialization:

Step 1: assign the parameters of AIS such as pop_{size} (population of antibody), $\max_{\text{gen}}$ (stop criterion), N_c (number of selected antibodies to clone).

Step 2: generate population of antibodies and evaluate their costs and affinity measures and sort them according to their affinity measure, gen=0.

Main loop:

For each generation do:

Step 3: select N_c antibodies from the highest affinity antibodies of population to clone.

Step 4: cloning according to affinity measures.

For each generated clone do:

Step 5: mutation according to affinity measures.

Step 6: determine the affinity measures of mutants.

Step 7: select N_c antibodies of mutants with the highest affinity measures and replace them with N_c antibodies of population which have the lowest affinity measures, gen=gen+1.

Step 8: repeat the whole steps until the stop cri-

terion is satisfied (gen> $\max_{gen}$).

4.1.1 Basic concepts of the algorithm

4.1.1.1 Solution encoding

The representation of each solution is called solution encoding. According to the main purpose of this paper, the objective is to determine the sequence of jobs and robots moves, assignment of jobs to batches and selection of optimal due dates. Each solution consists of the sequence of jobs and robots moves and batching. Also, to assign optimal due dates, each solution considers them in calculating the objective function. According to the above descriptions, representation of solution is as follows:

Each solution is defined as a cell which has n arrays (n is equal to the number of job). Each array consists of two-dimension matrices which have 4 rows and s columns (s is equal to the number of stage). The first row indicates number of operations which is performed by robots, the second row indicates number of jobs, the third row indicates number of stages and the last row indicates number of batches.

For instance, consider four jobs, four stages and two batches; the solution is defined as a cell with 4 arrays that each array is as Table 1.

To assign due dates for each job, the lemma which is presented by Shabtay [20] is used. For jth job, tardiness penalty weight (β_j), due date penalty weight (γ_j), delivery time (DT_j) and acceptable lead time (A_j) are considered. The optimal due date assignment policy is as follows:

For jth job if $DT_j \leq A_j$ then set $dd_j^* = DT_j$ otherwise if $Ct_j \geq Cg_j$ set $dd_j^* = A_j$, and if $Ct_j < Cg_j$ set $dd_j^* = DT_j$.

4.1.1.2 Affinity function

The affinity function is equivalent to fitness function of genetic algorithm. This function calculates the affinity value of each antibody, defined as:

$$\text{Affinity}(z) = \frac{1}{Total\ Cost(z)}$$

Where z is a given antibody. The antibody that has the lowest total cost will have the highest affinity value. Since the cloning of an antibody is proportional to its affinity, the antibody with the lowest total cost is cloned more than the antibody with the highest total cost.

4.1.1.3 Cloning

After calculating affinity value of each antibody and sorting them according to affinity measure, N_c antibodies are selected for cloning. Cloning the selected antibodies is done according to affinity measure. It means that the highest affinity measure antibody is cloned more than the lowest affinity antibody. Hence each antibody of N_c antibodies is cloned among $N_c - k + 1$, where k is the number of selected antibodies to clone after sorting.

4.1.1.4 Mutation

For each antibody two mutation procedures are applied: swap and reversion which are described as below:

4.1.1.4.1 Swap: for an antibody, let i and j be two randomly selected arrays of an antibody's cell. Replace i with j.

4.1.1.4.2 Reversion: for an antibody, let i and j be two randomly selected arrays of an antibody's cell. Reverse arrays between i and j.

4.2 Tabu search algorithm

Tabu search is a meta-heuristic neighborhood search methodology which is introduced by Glover (1986). This method acts as a local search procedure. The main idea of this technique is to apply a move to the best solution in the neighborhood. This algorithm attempts to avoid cycling and escape from local optimality by forbidding reverse moves, hence it maintains a list of prohibited moves called a tabu list. This algorithm is able to maintain a short-term memory function which determines for how long a tabu restriction is enforced. It means that the forbidden movements can be overridden their prohibition when a certain criterion (aspiration criteria) is satisfied. The aspiration criteria revoke tabu restriction if at any iteration of search process, there is a tabu move which can generate a solution that is better than the best solution found so far. The basic concepts of TS are described as follows:

At first, this algorithm starts searching process from an initial solution. Then, a set of solutions which is called neighborhood is generated by applying subset of moves to a current solution. At each step, to find appropriate neighbor, the neighborhood of current solution is searched. The move which leads the solution to the appropriate neighbor is determined as forbidden move and attributes of this move are recorded on tabu list for a chosen span of time which is called tabu tenure. Nevertheless, a forbidden move can be performed if aspiration function evaluates it as sufficiently profitable [17]. Also, another effective method of TS is long term memory function which attempts to achieve intensification and diversification. The proposed algorithm is summarized below:
Step 1: generate a feasible solution (x), evaluate it $(f(x))$.
Step 2: $x^* = x$, $f^* = f(x)$ where x^* is the best solution found so far.
Step 3: $it = 0$ (iteration counter), $TL = \emptyset$ (set of tabu movements).
Step 4: apply allowable moves to current solution and generate a set of neighborhood solutions, $N(x)$.
Step 5: if $N(x) - TL = \emptyset$ go to step 2 otherwise $it = it + 1$.
Step 6: select $n_{best} \in N(x) - TL$ which n_{best} is best solution of $n \in N(x) - TL$, denoted as: $(n_{best}(x) = opt(n(x) : n \in N(x) - TL)$: $opt()$ chose a solution which is generated from best movement).
Step 7: $x \leftarrow n_{best}(x)$, If $f(x) < f^*(x)$ then $x^* \leftarrow x$.
Step 8: check the stop criteria, if the stop criteria are not satisfied, update TL and return step 2.

The encoding of solution was explained in section 4.1.1.1 Also, the procedures which were explained in section 4.1.1.4 are used for generating neighborhood solutions.

5 Computational results

To illustrate the effectiveness of the proposed algorithms, some test problems of Elmi's paper [6] are considered which their defined parameters are reported in Table 2. The effectiveness of algorithms is evaluated in comparison with the results obtained from SA algorithm for makespan objective. Also, to access the performance of the proposed algorithms, the computational experiments have been carried out on a set of randomly generated instances which their defined parameters are reported in Table 3. The performances of the proposed algorithms are evaluated via comparing the results obtained by them.

In order to obtain the results of computational experiments and test problems, at first, parameter calibration should be done. More details are discussed as follows:

The parameters designing has significant impact on the efficiency of the proposed algorithms, hence, to design parameters of proposed algorithms, Taguchi method is used in this research. In order to calibrate parameters, three levels of parameters have considered for different problem sizes which are categorized by a length of solution in three intervals, denoted as $solution_{length} = j(s-1)$ as shown in Table 4. These intervals categorize problems according to three groups where the proposed algorithms parameters and their considered level are shown in Tables 5 and 6.

Since by increasing number of parameters, a large number of experiments have to be carried out, Taguchi method is applied to deal with this problem. In order to solve this problem, this method uses a special design of orthogonal arrays to study the entire parameters space with only a small number of experiments. As recommended in Taguchi method, the experimental results are determined in a signal-to-noise(S/N) ratio to find appropriate levels of parameters. The signal-to-noise(S/N) ratio measures the quality characteristics deviation from the desired value.

In order to employ the orthogonal arrays of Taguchi and the signal-to-noise ratio, it is necessary to know the number of parameters and the number of levels; hence, the L_9 orthogonal array is selected to study the entire parameters space. Therefore to find appropriate level of parameters, nine experiments are considered. To reduce the effect of stochastic nature of proposed algorithms, they are implemented five times for each problem and the average objective values of five runs are transformed to S/N ratio. The appropriate levels of parameters have been specified according to

the S/N ratios and are reported in Tables 7 and 8.

5.1 Comparison between proposed algorithms and SA

After determining appropriate levels of parameters, to illustrate the effectiveness of the proposed algorithms, 15 test problems of Elmi's paper [6] are considered and have been solved 5 times. The average makespan values over five runs are reported in Table 9. Also, the results obtained by SA algorithm of Elmi's paper [6] are reported in this table.

These algorithms were coded in MATLAB and run on Intel(R) Core(TM) i5-M460 2.53GHz 4.00GB-RAM.

To indicate the effectiveness of the proposed algorithms, percentage of reduction in average makespan for AIS and TS compared to SA has been denoted as GAP in below and shown in Table 10.

$$GAP = \left(\frac{SA_{makespan} - Proposed\ algorithm_{makespan}}{SA_{makespan}}\right) \times 100$$

Table 9 indicates that both proposed algorithms have reduced the objective by $\overline{GAP}_{AIS} = 19.96\%$ and $\overline{GAP}_{TS} = 18.34\%$. Consequently, the production rate is increased by $PR_{AIS} = 15.97\%$ and $PR_{TS} = 14.97\%$ as below:

$$PR = \left[\frac{\overline{GAP} \times 10^{-2}}{1 - \overline{GAP} \times 10^{-2}}\right] \times 100$$

With regard to the obtained results, the effectiveness of proposed algorithms is proved.

5.2 Comparison between AIS and TS

As mentioned previously, to investigate the performance of the proposed algorithms, the computational experiments have been carried out on a set of randomly generated instances. The obtained result from AIS algorithm is compared with the ones from TS algorithm. In order to compare the algorithms, we used the procedure which is used in Ramazani's paper [18]. This procedure is as follows:

At first the proposed algorithms are run for each problem which is generated randomly by parameters in Table 3, reported in Table 11. The main idea of this procedure states that both algorithms are run for the problem at the same time. Since the obtained results indicate that the run time of TS is greater than the run time of AIS, if TS is run at the same duration as AIS run time, it will cause damage to convergence of TS. Hence in this research, we used the average relative percentage deviation (ARPD) as the performance metric to evaluate the algorithms and their CPU times to compare them in terms of performance. In order to quantify the employed ARPD, the average objective value of five run is calculated for each generated problem which is denoted as Alg_{sol}. The best obtained solution of each generated problem among the two proposed algorithms is calculated which is denoted as $Best_{sol}$. Subsequently the RPD and ARPD are calculated as follow:

$$RPD = \frac{Alg_{sol} - Best_{sol}}{Best_{sol}} \times 100$$

$$ARPD = \frac{\sum_{i=1}^{Number\ of\ Run} RPD_i}{Number\ of\ Run}$$

The obtained results are reported in Table 12 and Figs. 2 and 3.

According to Table 12 and Fig. 1, AIS algorithm achieved 72.22% better solutions in terms of quality in quite shorter computing times for these problems. Therefor the results show that AIS performs well for this problem. In order to determine the reaction of proposed algorithms against changing the number of robots, stages and jobs, we considered the effects of different number of robots, stages and jobs on the objective value which are shown in Table 13. According to the results of Table 13, the reduction in objective value for each problem as the number of robots increases, is indicated in Fig. 4 and 5 and the rise in objective value for each problem as the number of stages increases, is indicated in Fig. 6 and 7.

6 Conclusion

This paper scrutinizes the blocking hybrid flow shop robotic cell scheduling (BHFS-RCS) prob-

lem considering unrelated parallel machines, multiple part types and machine eligibility constraints. Objective function was considered to find sequence of jobs and robots moves and select optimal due dates which minimizes earliness, tardiness, due date assignments and delivery costs which is completely a new idea in BHFS-RCS realm. In order to solve this problem in small sizes, a mixed-integer linear programming (MILP) model is proposed. With regard to the existing literature in the context of robotic cell scheduling, AIS and TS algorithms have not been proposed to deal with the addressed problem. Hence for solving the model in real size problems two meta-heuristic algorithms, namely artificial immune system and tabu search algorithms are proposed. In addition, to evaluate the effectiveness of the proposed algorithms, 15 test problems of Elmi's paper [6] are considered to compare proposed algorithms with SA algorithm of Elmi's paper and to evaluate the performance of proposed algorithms, 18 test problems are generated randomly to compare AIS with TS. As a future study regarding real world cases of BHFS-RCS, in order to increase the applicability of the proposed approach, considering constraints such as machine breakdowns, sequence dependent set up times and preemptions is recommended. Furthermore, the proposed problem can be investigated in other scheduling systems such as job shops or open shops.

References

[1] K. Arviv, H. Stern, Y. Edan, Collaborative reinforcement learning for a two-robot job transfer flow-shop scheduling problem, *International Journal of Production Research* 54 (2016) 1196-1209.

[2] G. D. Batur, S. Erol, O. E. Karasan, Robot move sequence determining and multiple part-type scheduling in hybrid flexible flow shop robotic cells, *Computers & Industrial Engineering* 100 (2016) 72-87.

[3] Ü. Bilge, G. Ulusoy, A time window approach to simultaneous scheduling of machines and material handling system in an FMS, *Operations Research* 43 (1995) 1058-1070.

[4] J. Carlier, M. Haouari, M. Kharbeche, A. Moukrim, An optimization-based heuristic for the robotic cell problem, *European Journal of Operational Research* 202 (2010) 636-645.

[5] A. Che, V. Kats, E. Levner, An efficient bicriteria algorithm for stable robotic flow shop scheduling, *European Journal of Operational Research* 260 (2017) 964-971.

[6] A. Elmi, S. Topaloglu, A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots, *Computers & operations research* 40 (2013) 2543-2555.

[7] A. Elmi, S. Topaloglu, Multi-degree cyclic flow shop robotic cell scheduling problem with multiple robots, *International Journal of Computer Integrated Manufacturing* 30 (2017) 805-821.

[8] A. Elmi, S. Topaloglu, Multi-degree cyclic flow shop robotic cell scheduling problem: Ant colony optimization, *Computers & Operations Research* 73 (2016) 67-83.

[9] A. Elmi, S. Topaloglu, Scheduling multiple parts in hybrid flow shop robotic cells served by a single robot, *International Journal of Computer Integrated Manufacturing* 27 (2014) 1144-1159.

[10] O. Engin, A. Döyen, A new approach to solve hybrid flow shop scheduling problems by artificial immune system, *Future generation computer systems* 20 (2004) 1083-1095.

[11] H. N. Geismar, M. Dawande, C. Sriskandarajah, Robotic cells with parallel machines: Throughput maximization in constant travel-time cells, *Journal of Scheduling* 7 (2004) 375-395.

[12] H. N. Geismar, M. Dawande, C. Sriskandarajah, Throughput optimization in constant traveltime dual gripper robotic cells with parallel machines, *Production and Operations Management* 15 (2006) 311-328.

[13] J. Hurink, S. Knust, A tabu search algorithm for scheduling a single robot in a job-shop environment, *Discrete applied mathematics* 119 (2002) 181-203.

[14] J. Hurink, S. Knust, Makespan minimization for flow-shop problems with transportation times and a single robot, *Discrete Applied Mathematics* 112 (2001) 199-216.

[15] M. Kharbeche, J. Carlier, M. Haouari, A. Moukrim, Exact Method for Robotic Cell Problem, *Electronic Notes in Discrete Mathematics* 36 (2010) 859866.

[16] W. Lei, H. Manier, M. A. Manier, X. Wang, A hybrid quantum evolutionary algorithm with improved decoding scheme for a robotic flow shop scheduling problem, *Mathematical Problems in Engineering* 5 (2017) 11-27.

[17] E. Nowicki, C. Smutnicki, A fast tabu search algorithm for the permutation flow-shop problem, *European Journal of Operational Research* 91 (1996) 160-175.

[18] P. Ramezani, M. Rabiee, F. Jolai, No-wait flexible flowshop with uniform parallel machines and sequence-dependent setup time: a hybrid meta-heuristic approach, *Journal of Intelligent Manufacturing* 26 (2015) 731-744.

[19] S. P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, W. Kubiak, Sequencing of parts and robot moves in a robotic cell, *International Journal of Flexible Manufacturing Systems* 4 (1992) 331-358.

[20] D. Shabtay, Scheduling and due date assignment to minimize earliness, tardiness, holding, due date assignment and batch delivery costs, *International Journal of Production Economics* 123 (2010) 235-242.

[21] D. Shabtay, K. Arviv, Optimal robot scheduling to minimize the makespan in a three-machine flow-shop environment with job-independent processing times, *Applied Mathematical Modelling* 40 (2016) 4231-4247.

[22] D. Shabtay, G. Steiner, Two due date assignment problems in scheduling a single machine, *Operations research letters* 34 (2006) 683-691.

[23] A. Soukhal, P. Martineau, Resolution of a scheduling problem in a flowshop robotic cell, *European Journal of Operational Research* 161 (2005) 62-72.

[24] N. H. Tuong, A. Soukhal, Due dates assignment and JIT scheduling with equal-size jobs, *European Journal of Operational Research* 205 (2010) 280-289.

[25] X. Yu, M. Gen, Introduction to evolutionary algorithms, *Springer Science & Business Media*, (2010).

[26] S. S. Zabihzadeh, J. Rezaeian, Two metaheuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time, *Applied Soft Computing* 40 (2016) 319-330.

Javd Rezaeian received his Ph.D. in Industrial Engineering from Mazandaran University of Science and Technology (MUST) in 2010. He is currently an Associate Professor and a faculty member of the department of Industrial Engineering at MUST. His Research interests are applications of project management, sequencing and scheduling, facility design, and metaheuristics.

Negar Derakhshan received her B.E. in Industrial Engineering from Tabriz University in 2011, and M.E. in Industrial Engineering from Mazandaran University of Science and Technology in 2015. She has been working in Iran Rail Industries Development Company since she was graduated as a master of engineering. Her main fields of work and study are manufacturing of different type of turnouts for railways, metros, and

mine industries, as well as robotic scheduling, and multi objective evolutionary algorithms.

Iraj Mahdavi received his B.E. degree in Industrial Engineering from Sharif University of Technology and M.E. in Industrial Engineering from Iran University of Science and Technology, and Ph.D. in Industrial Engineering from Govind Ballabh Pant University of Agriculture and Technology. He is currently the president of Mazandaran University of Science and Technology and a Distinguished Professor of Industrial Engineering. His research interests include cellular manufacturing, supply chain, production planning, scheduling, and soft computing.

Reza Alizadeh Foroutan received his B.E. and M.E. in Industrial Engineering from Mazandaran University of Science and Technology in 2009 and 2017 respectively. Having been working since graduated as a bachelor of engineering, he currently works as a project manager in the field of information technology and is an independent researcher. His research interests are in the areas of scheduling, logistics and supply chain, optimization problems, and soft computing.