E-ISSN: 2676-6167 | P-ISSN: 2538-5100

Journal of Applied Research on Industrial Engineering

www.journal-aprie.com

J. Appl. Res. Ind. Eng. Vol. x, No. x (xx) x–x.

Paper Type: Original Article

A Bi-Objective Optimal Task Scheduling Model for Two-Machine Robotic-Cell Subject to Probable Machine Failures

Bahareh, Vaisi [1,*], Hiwa, Farughi[2], Sadigh, Raissi[3], Heibatolah, Sadeghi[4]

1 Young Researchers and Elite Club, South Tehran Branch, Islamic Azad University, Tehran, Iran; st_b_vaisi@azad.ac.ir;
2 Department of Engineering, University of Kurdistan, Sanandaj, Iran; h.farughi@uok.ac.ir;
3 School of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran; Raissi@azad.ac.ir;
4 Department of Engineering, University of Kurdistan, Sanandaj, Iran; h.sadeghi@uok.ac.ir;

Citation:

LastName, (Abreviation of FirstName)., & LastName, (Abreviation of FirstName. (Date). Paper Title. *Journal of applied research on industrial engineering* , *Volume* (Issue), PP.

Received: Reviewed: Revised: Accept:

Abstract

In this study, we model a stochastic scheduling problem for a robotic cell with two unreliable machines susceptible to breakdowns and subject to the probability of machine failure and machine repair. A single gripper robot facilitates the loading/unloading of parts and cell-internal movement. Since it is more complicated than the other cycles, the focus has been on the S_2 cycle as the most frequently employed robot movement cycle. Therefore, a multi-objective mathematical formulation is proposed to minimize cycle time and operational costs. The ε-constraint method is used to solve small-sized problems. Non-dominated sorting genetic algorithm II (NSGA-II), is used to solve large-sized instances based on a set of randomly generated test problems. The results of several Test problems were compared with those of the GAMS software to evaluate the algorithm's performance. The computational results indicate that the proposed algorithm performs well. Compared to GAMS software, the average results for maximum spread (D) and non-dominated solutions (NDS) are 0.02 and 0.04, respectively.

Keywords: Breakdowns; Identical parts; NSGA-II; Probable failures; Robotic cell; Scheduling.

1 | Introduction

Flexible manufacturing systems (FMS) play an essential role in production systems and promptly respond to customer demands [1]. In such systems, robots are typically responsible for picking up products and loading/unloading machines; consequently, robots can facilitate the process and improve system productivity. A robotic cell is a type of FMS consisting of m computerized numerical control (CNC) machines; Some robotic cells also have an input and output buffer. In robotic cell problems, the primary focus of research is on scheduling robot tasks. Scheduling optimization, which improves the system's productivity when a manufacturing system must deal with uncertainty, is essential.

Corresponding Author: st_b_vaisi@azad.ac.ir

Scheduling is one of the most critical issues in all systems that optimize one or more objectives by considering the resource and operation constraints [2]. This issue's application in various domains, such as production and service systems, can assist the systems in achieving their desired performance goals [3]. As we can see in service systems such as hospitals, several factors can lead to a stop in operations, and proper scheduling that considers resources and constraints increases efficiency [4]-[5]. Machine breakdowns and repair times have been relaxed in the scheduling optimization process of robotic cells so far. Therefore, the current study proposes scheduling a two-machine robotic cell that confronts breakdowns. The robot performs under the S_2 cycle as the most commonly used robot's movement cycle. We use the ε-constraint method to solve the small-sized and NSGA-II to solve the large-sized problems.

The review of the literature is summarized in the following section. Section 3 defines the problem and presents the mathematical model used to model it. The solution approach is presented in Section 4, followed by some numerical examples, sensitivity analysis of results, and discussion based on the model in Section 5. Finally, Section 6 reports on the paper's conclusions.

2 | Literature Review

In most previous studies on robotic manufacturing cells, the scheduling problem considers a single criterion. Their research's most important objective functions were minimizing cycle time and maximizing the cell's throughput. Such as the papers cited in [6]-[15]. Hoogeveen's [16] survey of multi-criteria scheduling was published. The problem of multi-objective scheduling in robotic cells has been studied by [17]-[30], among others.

Given the importance of completing different tasks on time, deterioration and delays between tasks incur enormous costs. Consequently, maintenance is a crucial aspect of industrial environments. Although there have been numerous studies on deterministic robotic cells, the issue of determining an unreliable robotic cell in both machine breakdowns distributed according to an exponential distribution and stochastic processing time (due to the probability of repair time) remains unsolved. Stochastic models incorporate uncertainty and utilize probability distributions in which the data are either known or can be estimated.

Recent studies have shown that stochastic factors, such as machine breakdown and uncertain repair time, significantly impact the scheduling in actual production environments. Considering a multilevel assembly system with multiple sublevel components, [31] stated that it would be impossible to complete the items on time due to random machine breakdowns. They then proposed a mathematical model incorporating the uncertainty of lead time. Additionally, [32] stated that their operating costs might increase due to using tools and machines, thereby increasing the system's expenses. Utilizing preventive maintenance is proposed as a method for reducing operational costs. The history of the study of stochastic robotic cells is as follows.

1) Some previous studies focus on robotic cells with random processing time, which can be referred to [33], [34], [35], and [13]. Shafiei-Monfared et al. [33] considered a robotic cell consisting of three machines and a robot in the center of the cell when a part processing time element is stochastic. Comparing the cycle times of a variety of scenarios in this robotic cell was an attempt to determine the productivity benefits of each. [34] presented the first analytic study of robotic cell operations where the process has a stochastic processing time, as is typical in the microlithography portion of semiconductor manufacturing. It was demonstrated how the proximity of the stochastic process to the bottleneck process influences throughput measurement in such cells. The robot's sequence time distribution function was identified and validated through simulation. In a different study, a robotic cell problem with variable processing times was formulated, and the effectiveness of heuristic and metaheuristic solution methods for optimizing output rate was demonstrated [35]. Tonke et al. [13] developed an online-offline scheduling approach based on the assumption of uncertain processing times to address real-world applications such as cluster tools in semiconductor manufacturing. Their research involved a dual-gripper robotic cell problem with pick-up constraints.

2) According to other studies like [36]-[39], [28], and [30], robotic cells operate under a production system with machine failures and repairs. Savsar and Aldaihani [36] developed a model to analyze performance measures (PFM) of a flexible manufacturing cell (FMC) consisting of two machines and a robot under various operational conditions, including machine failures and repairs. The model was based on Markov processes and determined closed-form probabilities of system states for calculating PFMs. In a separate study [37], fault-tolerant conditions were incorporated into the model, allowing the FMC to operate in a degraded state. A Markovian model was developed to determine the system's dependability and productivity under various operational conditions. In a similar study by [38], the Markov chain model was developed for both single- and dual-machine FMCs. The model was subsequently generalized to FMC with n machines. Researchers [28], [30], and [39] investigated random failures in robotic cells with two and three machines. In recent researches conducted by [39]-[42] and [30], the robotic cell produces a variety of parts in an uncertain environment.

This study addresses a stochastic issue for an unreliable two-machine robotic cell when it considers the probability of machine failure and its impact on cycle time uncertainty. A single gripper robot is utilized to load and unload identical parts. Here, the authors focused on the S_2 movement cycle, which is more complicated than the other cycles and is the most commonly used movement cycle for robots.

3 | Problem Definition and Modeling

One type of product flows over only one machine in the manufacturing cell, but there are two identical CNC machines, neither of which has operational priority. Typically, in a 2-machine cell, there are three possible robot cycle options for part displacement: S_1, S_2 and $S_{12}S_{21}$ [6, 40]. As mentioned previously, the scope of this paper is restricted to the S_2 cycle.

In the S_2, a robot initially takes place before the Input Buffer (IB). Then the following operations are followed sequentially by the robot. 1) the robot picks up a part, 2) moves to the first machine (M1), 3) loads M1, 4) the robot moves to the second machine (M2), 5) waits for the previous process to be completed on the part (if it is needed), 6) robot unloads the part from M2, then, 7) transfers the product to the Output Buffer (OB), 8) loads the OB, 9) robot returns to M1, 10) if it is needed the robot waits until the completion of the process, 11) unloads from M1, 12) transports the part to the M2, 13) loads the part on the M2 and finally 14) robot turns back to the IB [6]. $A_{01}\ A_{23}\ A_{12}$ infers the sequence of activities in the S_2 cycle as mentioned; A_{pq} is the robot's activity sequence from station p to station q for $p = 0, 1, 2$ & $q = 1, 2, 3$; see [6] and [40]. A typical linear two-machine robotic cell is shown in Figure1.

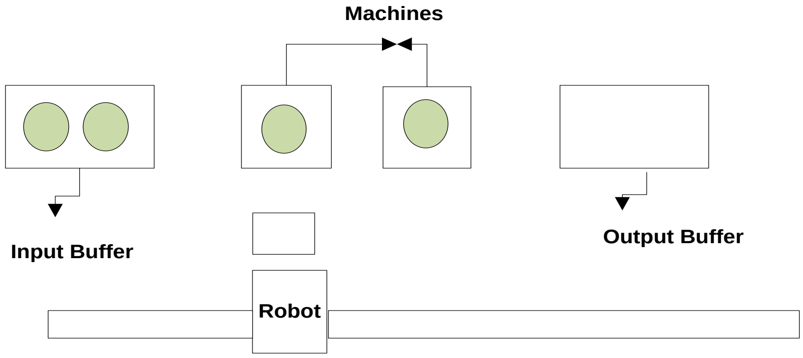

Fig. 1. Linear two machines robotic cell.

3.1 | Assumptions

In addition to the assumptions in [28], the basic assumptions for this study are:

1. Failure in the machines is probable.
2. Some repairs performed to the machine per part's entrance to the cell follow an exponential distribution. The time to repair each machine is equivalent to its processing time.
3. To produce each part, multiple operations are required. Some of the operations are done on the first machine and the rest is done on the second machine.
4. Existence of probability distribution in repair time may face the processing time with uncertainty. So, it is assumed that; the machines are unreliable and there is a possibility of failure during the operation. It is also assumed that the number of repairs performed per machine for each part follows the Exponential distribution. By arriving the part into the cell, considering the probability of machine failure, there are uncertain processing times that complicate the analysis and modeling of the robotic cell. The following model is developed to analyze the desired robotic cell. The layout of the assumed robotic cell was based on Fig.1 and [21].

3.2 | Notations

The following parameters and variables are used in the proposed mathematical model.

$C_0{}^k$	Machining cost per kth part ($/min).
C_r	Repair cost for each breakdown ($/min): without setup costs
C_p	Preventive maintenance cost ($/min)
C_{TOOL}	Cost of tool ($/tool): Tools replacement prohibited in an operating cycle
λ	Failure rate: follows the Exponential distribution
μ	Repair rate: follows the Exponential distribution
t_i	Processing time of operation i (min)
$t_{r,j}$	Duration of a repair visit for machine j (min)
W_{R_j}	Duration of maintenance in machine j (min)
W^k	Robot's waiting time in a cycle for the k^{th} part fed to the cell (min)
F^k	Total cost for the k^{th} part fed to the cell ($)
$T_{S_2}{}^k$	The partial cycle time of S_2 for the k^{th} part fed to the cell (min)
$X_j{}^k$	Number of repairs performed to the machine j for the k^{th} part entered to the cell (integer random variable, Exponential distribution (meaning $X_j^k \geq 1$)
a	The processing time for a part on the 1^{st} machine
b	The processing time for a part on the 2^{nd} machine
ϵ	Loading/unloading time

δ	Time is taken by a robot to move between two adjacent stations
$pr(TI_{\Theta} < TI < TI_{\Theta+1})$	The probability of breakdown occurrence for the time interval (TI) between TI_{Θ} and $TI_{\Theta+1}$

$$O_{ij} = \begin{Bmatrix} 1 & if\ Operation\ i\ is\ allocated\ to\ machine\ j, \\ 0 & therwise \end{Bmatrix}$$

3.3 | Modeling

Literature reveals that various studies, such as [43], have focused on minimizing the total production costs of machining, tooling, and maintenance. In many instances, tooling cost is considered a constant value and has no impact on the optimization process. In the present study, tooling costs are also constant. It was added to the formula to complete the concept. Total production cost for the k^{th}part fed to the manufacturing cell was defined as the first objective function while minimizing partial cycle time for the k^{th}part was considered as the second objective function. Equations (1) and (2) represent the preferred objective functions.

In the proposed model, the constraints for robotic cell scheduling are equations (3), (4), (5), and (8). These equations are derived from the robot's move cycle definitions and redefined based on the assumed problem. The time between two consecutive repairs for the first machine is a and for the second machine, b. Each machine's maintenance time is represented by Equation (6). The failure rate per machine for each model run is calculated using an Exponential distribution. Equations (7) and (9) are decision variables related to the allocation of operations to machines. The formulation example follows.

$$Min\ F^k = \left(\sum_{j=1}^{2} \sum_{i=1}^{n} C_0{}^k\, t_i O_{ij} + C_r\, t_{r,j}\, pr(TI_{\Theta} < TI < TI_{\Theta+1}) \right) + C_{TOOL} \tag{1}$$

$$Min\ T_{S_2}{}^k = 6 \in +8\delta + W^k \tag{2}$$

s.t.

$$a = \left(\sum_{i=1}^{n} t_i O_{i1} \right) + W_{R_1} \tag{3}$$

$$b = \left(\sum_{i=1}^{n} t_i O_{i2} \right) + W_{R_2} \tag{4}$$

$$W^k = Max \begin{Bmatrix} 0, aX_1{}^k - \left[2 \in +4\delta + W_{R_1}\right], \\ bX_2{}^k - \left[2 \in +4\delta + W_{R_2}\right] \end{Bmatrix} \tag{5}$$

$$W_{R_j} = \sum_{i=1}^{n} t_{r,j}\, pr(TI_{\Theta} < TI < TI_{\Theta+1}) O_{ij} \quad for\ j=1,2 \tag{6}$$

$$O_{i1} + O_{i2} = 1 \tag{7}$$

$$W^k \geq 0 \tag{8}$$

$$O_{ij} \in \{0,1\} \tag{9}$$

4 | Solution approach

Bi-objective optimization problems aim to identify a set of Pareto optimal solutions. This research uses the evolved ε-constraint method for small-scale problems. Recently, [44] reviewed the application of optimization tools in robotic systems and revealed that the authors of nearly half of the research papers published between 2005 and September 2021 had used heuristic/metaheuristic algorithms to optimize robotic manufacturing system problems. Consequently, a well-known multi-objective meta-heuristic approach, NSGA- II, is utilized to solve the bi-objective model in the current study for large-scale problems. This algorithm is one of the most popular multi-objective optimization algorithms. After presenting the first version of this algorithm in 1995, its developers, the most significant among whom is Debb, presented the second version, NSGA-II, in 2002. Figure 2 shows the Pseudo code of the NSGA-II.

1. Create:
   Population size= P_t

   Repeat for a maximum number of Iterations

   $t = 0$
2. Generate child population = Q_t
   Apply:
   -Binary Tournament selection
   - One-point Crossover and Probable Mutation
3. Combine P_t and Q_t to create a new population called Npop
4. Assign rank for each solution based on the non-domination sorting process
5. Create next-generation, (P_t), based on the lowest obtained ranks and highest Crowding Distance
6. Check the stopping criterion.
   While t < Maximum number of Iterations, do:
   If Yes (Go to step 7)/ If No $t = t + 1$ *and* (Go to step 2)

7. End of the algorithm.

Fig. 2. The NSGA-II Pseudo code.

4.1 | Crowding Distance Computation

The following two criteria determine the measures for better solutions:

- Rank measure

The solution with the lower non-domination rank is preferred between two alternatives with different ranks. Alternatively, if both points belong to the same front, the point located in a region with fewer points is preferred [45].

- Crowding distance

In instances where two selected particles occupy the same rank (both on the same side), the Crowding Distance criterion is applied, as explained below.

For particles 2 to $n-1$, the crowding distance, $I(d_k)$, is calculated based on equations (10) and (11).

$$CD_K = I(d_k)_1 + \ldots + I(d_k)_m \quad (10)$$

$$I(d_k).m = \frac{I(k+1).m - I(k-1).m}{f_m^{\max} - f_m^{\min}} \quad (11)$$

The equation (11) represents the crowding distance for the objective function m. Therefore, $I(d_k)$ must be calculated and summed for all objective functions, as specified in equation (10). After calculating the crowding distance (CD), the particle with the highest crowding distance is selected, [46]- [47].

4.2 | Solution Representation

The main objective is to determine the assignment of operations to machines and to plan the arrival of parts and their processes in the robotic cell. For setting parameters in the experimental design, this study uses a certain amount of algorithm repetition as a stopping criterion. The experiments were designed using the Taguchi method to adjust the algorithm's parameters. Therefore, a Taguchi-based experiment on ten randomly generated Test problems was designed. Three experiment levels were chosen for each parameter, including Crossover rate (p_c) and Mutation rate (p_m), based on previous research and trial and error. The experiment levels of these parameters are displayed in Table 1.

Table 1. Levels of Taguchi experiment.

Parameters	Levels		
	Low	**Middle**	**High**
p_c	0.6	0.8	0.9
p_m	0.1	0.15	0.2

As a result of the experiments and the fact that smaller response values are taken into account, Table 1's middle level consists of appropriate combinations based on the average response factor.

Crossover=80%, Mutation=15%

The population size and stopping criteria must be modified to implement the algorithm. By increasing the population size, the algorithm searches for more points in the space, and the quality and distribution of the results improve; however, if the number of population members becomes ten times greater, the time or required memory to solve the problem will be 100. Therefore, the population size for the proposed algorithm is 50. By increasing the number of algorithm replications, the model is given sufficient time to be solved, resulting in better results for larger values of this parameter. However, it should be noted that increasing the number of algorithm replications also increases the elapsed time. The suggested number of replications for the NSGA-II algorithm is 500, and the algorithm stops upon reaching 500 replications. Therefore, the maximum number of iterations is established as a stopping criterion, and 500 iterations are set as the stopping criterion.

A 2-string chromosome is employed to represent the assignment of processing times of operations to the machines. The first string represents the processing times assigned to the *M1* and the second string is the processing times assigned to the second machine. There are i operations available for each part to be processed, so the probability density function of selecting operations is Uniform. Consequently, in these two strings, a number between 2 and i shows the equivalent processing time allocated to the associated machine. $t_i(j)$ means assigning the i^{th}processing time to machine j. Fig. 3 shows a chromosome with eight operations as an initial population sample.

t_1(1)	**t_2(2)**	**t_3(1)**	**t_4(1)**	**t_5(2)**	**t_6(1)**	**t_7(2)**	**t_8(2)**

Fig. 3. Forming an initial population.

Creating an initial population is the first step. In this study, the initial population is generated by generating 2-string chromosomes proportional to the size of the population. As stated previously, the 2-string chromosomes contain the processing times of operations required to produce i^{th}part. After forming the initial generation, individuals must be chosen to form the subsequent generation. Location of solutions in a Pareto front (lower fronts are superior) and Crowded Distance are the selection criteria (in the same lower fronts). The new generation should be formed by altering specific characteristics of the parents.

In designing this algorithm, One-point Crossover and Probable Mutation have been used. Fig. 4 represents the Crossover operation. The numerical values in Fig. 4, as an example, specify the value of processing times of operations to produce i^{th} part.

Forming the initial population for this problem (2-string chromosomes):

Parent 1	2.78	5.88	2.28	3.36	4.53	5.92	3.31	4.04
Parent 2	4.22	8.4	8.79	7.64	9.6	5.56	2.95	6.05

After One-point Crossover

Child 1	2.78	5.88	2.28	3.36	4.53	5.56	2.95	6.05
Child 2	4.22	8.4	8.79	7.64	9.6	5.92	3.31	4.04

Fig. 4. One-point Crossover.

5 | Results and Discussion

The proposed solution approach was tested on ten different Test problems. These Test problems are randomly generated by MATLAB R2016b and executed on an ASUS laptop with 8 GB of RAM and an Intel(R) Core (TM) i7-4500U processor running at 1.80GHz 2.40GHz. The Designated Test problems are listed in Table 2. Table 3 describes the parameters and defined values for the considered robotic cell.

Table 2. Designated Examples.

Test problem	Number of operations per part	Processing times
1	8	(2,5)
2	8	(2,10)
3	8	(2,15)
4	8	(2,30)
5	20	(2,5)
6	20	(2,10)
7	20	(2,15)
8	30	(2,5)
9	30	(2,10)
10	30	(2,15)

Table 3. Characteristics of required parameters.

Parameters		
$C_{TOOL}=5$	$\epsilon=2$	$\mu=2$
$C_o=100$	$\delta=0.5$	$\lambda=3$
$C_r=20$	$C_p=12$	$t_r=[2,3]$

The results of running the NSGA-II algorithm on the Test problems are presented in Table 4. It should be noted that for each Test problem, the algorithm has been executed five times, and Table 4 contains the best answers. Processing times have a uniform distribution within the specified range. In most Test problems, the distance between the upper and lower bounds of the objective functions increases as the number of part operations increases.

Table 4. Characteristics

Test problem	The lower bound of the first objective function	The Upper bound of the first objective function	The lower bound of the second objective function	The Upper bound of the second objective function
1	2265.34	2346.18	91.01	91.21
2	2474.10	2721.64	110.72	117.54
3	2309.42	2369.03	102.96	104.51
4	5452.48	6275.21	428.44	1002.29
5	7652.46	8153.35	909.26	1092.75
6	6345.30	6353.20	731.17	777.68
7	6316.70	7321.20	725.25	731.91
8	11566.08	12492.32	2153.21	2307.21
9	15698.68	17440.95	3486.69	3747.26
10	19225.42	21234.51	4275.92	4483.20

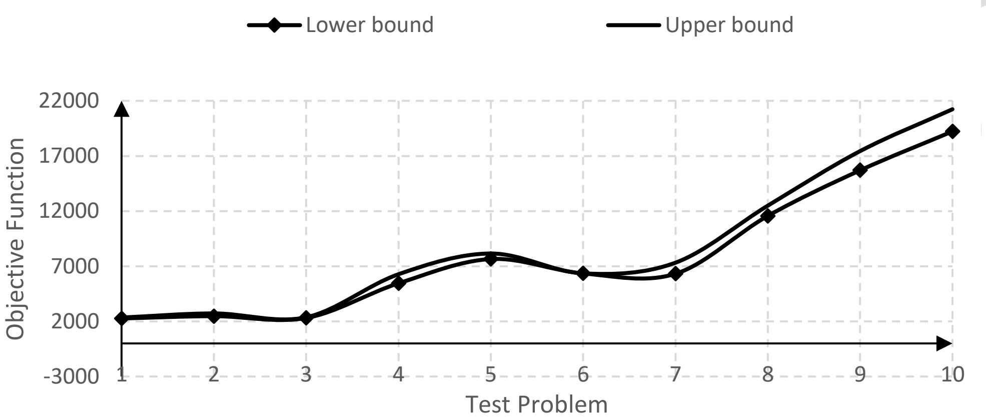

Fig. 5. Lower/Upper bound changes of the first objective function in the Test problems

Figures 5 and 6 demonstrate that the objective functions increase in value as the number of operations per part increases. In addition, the difference between the upper and lower bounds in the small-sized problems is not detectable, except for Test problem number 4. As the processing time increases in this Test problem, it affects the cycle time and total cost. The difference between the bounds is minor in instances with more than eight operations per part.

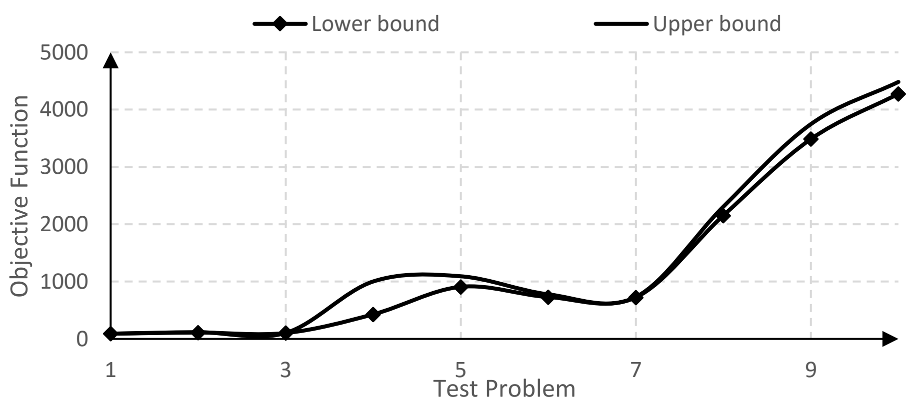

Fig. 6. Lower/Upper bound changes of the second objective function in the Test problems

Changes in the first objective function are compared to alterations in the processing time interval under the assumption of a constant number of operations. These variations are illustrated in Figures 7 to 9. Consequently, increasing the processing time interval causes the difference between the upper and lower total cost bounds to rise (first objective function).

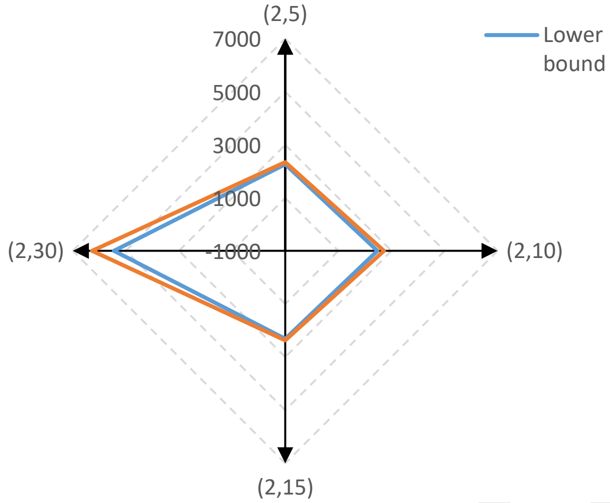

Fig. 7. Changes in the first objective function compared to changes in the processing time interval for eight operations (problems 1 to 4)

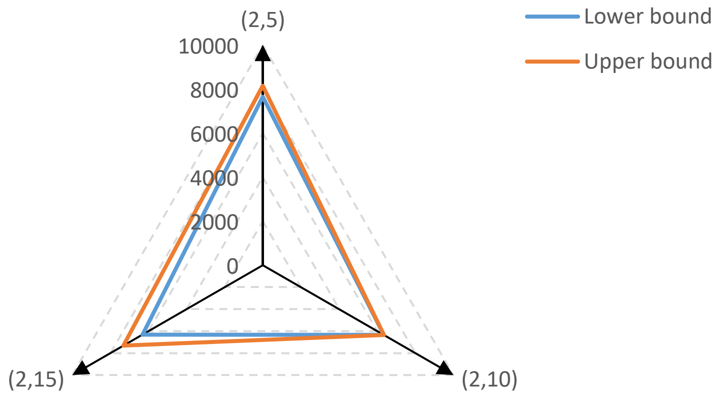

Fig. 8. Changes in the first objective function compared to changes in the processing time interval for 20 operations (problems 5 to 7)

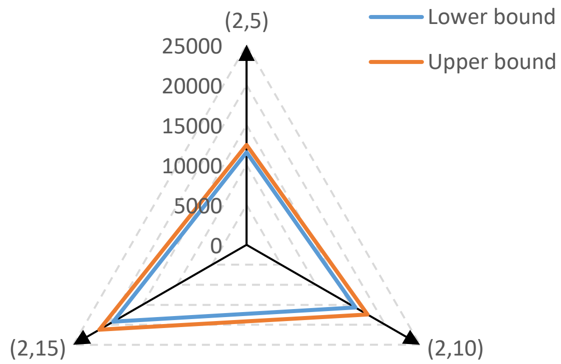

Fig. 9. Changes in the first objective function compared to changes in the processing time interval for 30 operations (problems 8 to 10)

5.1. Comparison of the algorithms

Different test problems of varying sizes are executed to compare the proposed algorithms. The performance of the algorithms is then evaluated using three standard evaluation indices (criteria): computational time, maximum spread (D), and non-dominated solutions (NDS). The size of the problem is the number of operations per part. The computational time is the average time required to provide a solution. Maximum spread evaluates the variety and distribution of Pareto front solutions using equation (12). The performance of an algorithm with a higher maximum spread is superior. Finally, NDS displays the number of non-dominated solutions obtained for each Test problem [48]-[49].

$$D = \sqrt{\sum_{m=1}^{M}\left(max_i\, f_m^i - min_i\, f_m^i\right)^2} \qquad (12)$$

Table 5 displays the mean values of the comparison metrics for each Test problem based on the GAMS and NSGA-II results.

Table 5. The result of the NSGA-II algorithm and GAMS for the proposed programming

Test problem	Number of operations per part	Processing times	GAMS results			NSGA-II results		
			NDS	*D*	*T(s)*	*NDS*	*D*	*T(s)*
1	8	(2,5)	9	1913.10	539	9	1913.10	42
2	8	(2,10)	12	1284.20	932	12	1284.20	85
3	8	(2,15)	17	2455.20	1450	17	2455.20	259
4	8	(2,30)	24	2884.40	4356	23	2884.40	587
5	20	(2,5)	29	12871.30	5124	27	12256	218
6	20	(2,10)	45	23105.38	9879	45	22890	759
7	20	(2,15)	49	40972.12	15491	46	39681	1412
8	20	(2,30)	58	52321.72	35270	56	50227	3104
9	30	(2,5)	35	17981.00	22872	32	17683	1862
10	30	(2,10)	56	40201.00	42501	52	39681	2412
11	30	(2,15)	67	82812.23	61731	60	80685	2810
12	30	(2,30)	76	145238.45	71025	73	142256	3654
13	50	(2,5)	49	46208.39	48654	44	45773	3940
14	50	(2,10)	__	__	*	59	55764	4105
15	50	(2,15)	__	__	*	65	85282	6120
16	50	(2,30)	__	__	*	74	149283	7345

According to Table 5, the ε-constraint method can achieve the Pareto optimal solution set for small-sized problems. However, the NSGA-II algorithm could obtain the Pareto optimal solution set for the first four Test problems; in the following four Test problems, with 20 operations per part, the NSGA-II results are very close to the results of the GAMS. Furthermore, the larger the size of the Test problems, the more the proposed solutions have computational time. Therefore, to solve the large-sized problems, the NSGA-II algorithm was applied.

Relative percentage deviation (RPD) is calculated for 12 samples of the Test problems using equation (13) to evaluate the results of NSGA-II. The average of RPD per index is determined for the algorithm, and according to the low value of RPD, this method applies to larger problems as well. Table 6 displays the results.

$$RPD = \left|\frac{Alg_{Sol} - Best_{Sol}}{Best_{Sol}}\right| \times 100 \qquad (13)$$

Table 6. The relative percentage deviation of the NSGA-II algorithm for the Test problems

Test problem	Number of operations per part	Processing time	RPD		
			NDS	*D*	$T(s)$
1	8	(2,5)	0	0	0.92
2	8	(2,10)	0	0	0.91
3	8	(2,15)	0	0	0.82
4	8	(2,30)	0.04	0	0.86
5	20	(2,5)	0.07	0.05	0.96
6	20	(2,10)	0	0.01	0.92
7	20	(2,15)	0.06	0.03	0.91
8	20	(2,30)	0.03	0.04	0.91
9	30	(2,5)	0.08	0.02	0.92
10	30	(2,10)	0.07	0.01	0.94
11	30	(2,15)	0.10	0.03	0.95
12	30	(2,30)	0.04	0.02	0.95
Average	-	-	**0.04**	**0.02**	**0.91**

6 | Conclusion

The proposed model was primarily concerned with minimizing the production cost and S_2 cycle time in a two-machine, identical-parts robotic manufacturing cell subject to breakdowns such as machine failures and repairs. The problem was formulated, and a well-known metaheuristic algorithm, NSGA-II, was used to solve this bi-objective model. The solution approach was evaluated on some randomly generated problems, and the results were presented as the upper and lower bounds for the two objective functions. Due to the insignificant difference between the upper and lower bounds, the mean value can represent the real-valued amount of the objective functions. The results showed the robustness of the model and the algorithm. Expanding this problem and working on multiple parts of robotic cells for future research are recommended.

Conflicts of Interest

All co-authors have read and approved the manuscript, and there are no financial conflicts to disclose. We certify that the submission is original and is not currently being reviewed by another publication.

References

[1] Farughi, H., Dolatabadiaa, M., Moradi, V., Karbasi, V., & Mostafayi, S. (2017). Minimizing the number of tool switches in flexible manufacturing cells subject to tools reliability using genetic algorithm. *Journal of Industrial and Systems Engineering*, Vol. 10, (special issue on Quality Control and Reliability), 17-33.

[2] Zanjani, B., Amiri, M., Hanafizadeh, P., & Salahi, M. (2021). Robust multiobjective hybrid flow shop scheduling. *Journal of applied research on industrial engineering*, 8(1), 40-55. https://dx.doi.org/10.22105/jarie.2021.252651.1202

[3] Rashidi, H., & Hassanpour, M. (2020). A deep-belief network approach for course scheduling. *Journal of Applied Research on Industrial Engineering*, 7(3), 221-237. https://dx.doi.org/10.22105/jarie.2020.243184.1185

[4] Hamid, M., & Tavakkoli-Moghaddam, R., Vahedi-Nouri, B., & Arbabi, H. (2020). A mathematical model for integrated operating room and surgical member scheduling considering lunch break. *International Journal of Research in Industrial Engineering*, 9(4), 304-312.

[5] Khalili, N., & Shahnazari Shahrezaei, P., & Abri, A. G. (2020). A multiobjective optimization approach for a nurse scheduling problem considering the fatigue factor (case study: Labbafinejad Hospital). *Journal of applied research on industrial engineering*, 7(4), 396-423. https://dx.doi.org/10.22105/jarie.2020.259483.1215

[6] Sethi, S. P., Sriskandarajah, C., Sorger, G., Blazewicz, J. & Kubiak, W. (1992). Sequencing of Parts and Robot Moves in a Robotic Cell, *International Journal of Flexible Manufacturing Systems*, 4(3), 331-358. https://doi.org/10.1007/BF01324886

[7] Fathian, M., Kamalabadi, I. N., Heydari, M., & Farughi, H. (2011). A Petri net model for part sequencing and robot moves sequence in a 2-machine robotic cell. *Journal of Software Engineering and Applications*, 4(11), 603. http://www.SciRP.org/journal/jsea

[8] Fathian, M., Kamalabadi, I. N., Heydari, M., Farughi, H., & Naseri, F. (2012). Developing Petri net model and meta-heuristic algorithms for cyclic scheduling in 2-machine robotic cells. *African Journal of Business Management*, 6(15), 5456-5466. DOI: 10.5897/AJBM11.2715

[9] Fathian, M., Nakhai Kamalabadi, I., Heydari, M., Farughi, H., & Naseri, F. (2013). Applying Metaheuristic Algorithms for Output Rate Analysis in Two-Machine Robotic Manufacturing Cells. *International Journal of Advanced Robotic Systems*, 10(169). https://doi.org/10.5772%2F56051

[10] Majumder, A., & Laha, D. (2016). A new cuckoo search algorithm for 2-machine robotic cell scheduling problem with sequence-dependent setup times. *Swarm and Evolutionary Computation*, 28, 131-143. https://doi.org/10.1016/j.swevo.2016.02.001

[11] Wang, Z., Zhou, B., Trentesaux, D., & Bekrar, A. (2017). Approximate optimal method for cyclic solutions in multi-robotic cell with processing time window. *Robotics and Autonomous Systems*, 98, 307-316. https://doi.org/10.1016/j.robot.2017.09.020

[12] Gultekin, H., Coban, B., & Akhlaghi, V. E. (2018). Cyclic scheduling of parts and robot moves in m-machine robotic cells. *Computers & Operations Research*, 90, 161-172. https://doi.org/10.1016/j.cor.2017.09.018

[13] Tonke, D., Grunow, M., & Akkerman, R. (2019). Robotic-cell scheduling with pick-up constraints and uncertain processing times. *IISE Transactions*, 51(11), 1217-1235. https://doi.org/10.1080/24725854.2018.1555727

[14] Bożejko, W., Pempera, J., Smutnicki, C., & Wodecki, M. (2020). Cyclic Scheduling in the Manufacturing Cell. *Modeling and Performance Analysis of Cyclic Systems* (pp. 49-62). Springer, Cham. https://doi.org/10.1007/978-3-030-27652-2_3

[15] Kim, H. J., & Lee, J. H. (2021). Scheduling of Dual-Gripper Robotic Cells with Reinforcement Learning. *IEEE Transactions on Automation Science and Engineering*, 19(2), 1120-1136. https://doi.org/10.1109/TASE.2020.3047924

[16] Hoogeveen, H. (2005). Multicriteria Scheduling. *European Journal of Operational Research*, 167(3), 592–623. https://doi.org/10.1016/j.ejor.2004.07.011

[17] Kayan, R. K. & Akturk, M. S. (2005). A New Bounding Mechanism for the CNC Machine Scheduling Problems with Controllable Processing Times. *European Journal of Operational Research*, 167(3), 624–643. https://doi.org/10.1016/j.ejor.2004.07.012

[18] Gurel, S. & Akturk, M. S. (2007). Considering Manufacturing Cost and Scheduling Performance on a CNC Turning Machine. *European journal of operational research*, 177(1), 325-343. https://doi.org/10.1016/j.ejor.2005.11.029

[19] Turkcan, A., Akturk, M. S., & Storer, R. H. (2007). Due Date and Cost-based FMS Loading, Scheduling, and Tool Management. *International Journal of Production Research*, 45(5), 1183–1213. https://doi.org/10.1080/00207540600559955

[20] Restrepo, I. M., & Balakrishnan, S. (2008). Fuzzy-based Methodology for Multiobjective Scheduling in a Robot-Centered Flexible Manufacturing Cell. *Journal of intelligent manufacturing*, 19(4), 421–432. https://doi.org/10.1007/s10845-008-0093-5

[21] Vaisi, B., Farughi, H., & Raissi, S. (2020). Multiobjective Optimal Model for Task Scheduling and Allocation in a Two Machines Robotic Cell Considering Breakdowns. *WSEAS Transactions on Information Science and Applications*, 17, 1-8. 10.37394/23209.2020.17.a025103-920

[22] Yildiz, S., Akturk, M. S., & Karasan, O. E. (2011). Bicriteria robotic cell scheduling with controllable processing times. *International Journal of Production Research*, 49(2), 569-583. https://doi.org/10.1080/00207540903491799

[23] Feng, J., Che, A., & Wang, N. (2014). Bi-objective cyclic scheduling in a robotic cell with processing time windows and non-Euclidean travel times. *International Journal of Production Research*, 52(9), 2505-2518. https://doi.org/10.1080/00207543.2013.849015.

[24] Ma, K., Yan, P., & Dai, W. (2016, June). A hybrid discrete differential evolution algorithm for dynamic scheduling in robotic cells. *In 2016 13th International Conference on Service Systems and Service Management (ICSSSM)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICSSSM.2016.7538453

[25] Abd, K., Abhary, K., & Marian, R. (2016). Multiobjective optimization of dynamic scheduling in robotic flexible assembly cells via fuzzy-based Taguchi approach. *Computers & Industrial Engineering*, 99, 250-259. https://doi.org/10.1016/j.cie.2016.07.028

[26] Mansouri, S. A., Aktas, E., & Besikci, U. (2016). Green scheduling of a two-machine flow shop: Trade-off between makespan and energy consumption. *European Journal of Operational Research*, 248(3), 772-788. https://doi.org/10.1016/j.ejor.2015.08.064

[27] Ghadiri Nejad, M., Shavarani, S. M., Vizvári, B., & Barenji, R. V. (2018). Trade-off between process scheduling and production cost in cyclic flexible robotic cells. *International Journal of Advanced Manufacturing Technology*, 96. https://doi.org/10.1007/s00170-018-1577-x

[28] Vaisi, B., Farughi, H., & Raissi, S. (2018). Bi-Criteria Robotic Cell Scheduling and Operation Allocation in the Presence of Break-downs. *International Journal of Industrial Engineering & Production Research*, 29(3), 343-357. DOI: 10.22068/ijiepr.29.3.343

[29] Wu, X., Yuan, Q., & Wang, L. (2020). Multiobjective Differential Evolution Algorithm for Solving Robotic Cell Scheduling Problem with Batch-Processing Machines. *IEEE Transactions on Automation Science and Engineering*. https://doi.org/10.1109/TASE.2020.2969469

[30] Vaisi, B., Farughi, H., & Raissi, S. (2020). Schedule-Allocate and Robust Sequencing in Three-Machine Robotic Cell under Breakdowns. *Mathematical Problems in Engineering*, 2020. https://doi.org/10.1155/2020/4597827

[31] Sadeghi, H., Makui, A., & Heydari, M. (2016). Multilevel production systems with dependent demand with uncertainty of lead times. *Mathematical Problems in Engineering*, 2016. https://doi.org/10.1155/2016/4967341

[32] Sadeghi, H. (2019). A forecasting system by considering product reliability, POQ policy, and periodic demand. *Journal of Quality Engineering and Production Optimization*, 4(2), 133-148. DOI: 10.22070/JQEPO.2020.5087.1123

[33] Shafiei-Monfared, S., Salehi-Gilani, K., & Jenab, K. (2009). Productivity analysis in a robotic cell. *International Journal of Production Research*, 47(23), 6651-6662. https://doi.org/10.1080/00207540802372298.

[34] Geismar, H. N., & Pinedo, M. (2010). Robotic cells with stochastic processing times. *IIE Transactions*, 42(12), 897-914. https://doi.org/10.1080/0740817X.2010.491505

[35] Al-Salem, M., & Kharbeche, M. (2017). Throughput optimization for the robotic cell problem with controllable processing times. *RAIRO-Operations Research*, 51(3), 805-818. https://doi.org/10.1051/ro/2016064

[36] Savsar, M., & Aldaihani M. (2008). Modeling machine failures in a flexible manufacturing cell with two machines served by a robot. *Reliability Engineering & System Safety*, 93 (10), 1551-1562. https://doi.org/10.1016/j.ress.2007.06.002

[37] Savsar, M. (2010). Reliability modeling of a manufacturing cell operated under degraded mode. *Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management (IEOM 2010)* (PP. 374-385). IEOM.

[38] Hamasha, M. M., Alazzam, A., Hamasha, S., Aqlan, F., Almeanazel, O., & Khasawneh, M. T. (2015). Multimachine flexible manufacturing cell analysis using a Markov chain-based approach. *IEEE Transactions on components, packaging and manufacturing technology*, 5(3), 439-446. https://doi.org/10.1109/TCPMT.2015.2394232

[39] Vaisi, B., Farughi, H., & Raissi, S. (2021). Utilization of Response Surface Methodology and Goal Programming based on Simulation in a Robotic Cell to Optimize Sequencing. *Journal of Quality Engineering and Management*, 10(4), 327-338. https://dorl.net/dor/20.1001.1.23221305.1399.10.4.5.7

[40] Vaisi, B., Farughi, H., & Raissi, S. (2018). TWO-MACHINE ROBOTIC CELL SEQUENCING UNDER DIFFERENT UNCERTAINTIES. *International Journal of Simulation Modelling (IJSIMM)*, 17(2), 284-294. https://doi.org/10.2507/IJSIMM17(2)434

[41] Zahrouni, W., & Kamoun, H. (2021). Scheduling in robotic cells with time window constraints. *European Journal of Industrial Engineering*, 15(2), 206-225.

[42] Lee, J. H., & Kim, H. J. (2022). Reinforcement learning for robotic flow shop scheduling with processing time variations. *International Journal of Production Research*, 60(7), 2346-2368. https://doi.org/10.1080/00207543.2021.1887533

[43] Selim Akturk, M., & Gurel, A. S. (2007). Machining conditions-based preventive maintenance. *International Journal of Production Research*, 45(8), 1725-1743. https://doi.org/10.1080/00207540600703587

[44] Vaisi, B. (2022). A review of optimization models and applications in robotic manufacturing systems: Industry 4.0 and beyond. *Decision Analytics Journal*, 100031. https://doi.org/10.1016/j.dajour.2022.100031

[45] Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II. In *International conference on parallel problem solving from nature* (pp. 849-858). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45356-3_83

[46] Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multiobjective problems*. Springer (pp. 79-104) New York.

[47] Kiyani Ghalehno, R., Niroomand, S., Didekhani, H., & Mahmoodirad, A. (2022). A multi-objective formulation for portfolio optimization of credit institutions branches: case study of Keshavarzi bank of Sistan and Baloochestan. *Journal of Decisions and operations research*, 299-315. doi: 10.22105/dmor.2021.257591.1260. (In Persian)

[48] Sadeghi, H., & Mahmoodi, A. (2022). Multiobjective inventory model for material requirements planning with uncertain lead-time. *International Journal of Operational Research*, 43(4), 391-415.

[49] Shoaee, M., & Samouei, P. (2021). A Cross-Dock Warehouse Layout Design Using Multi-Objective Gray Wolf Optimization Algorithm. *Journal of Decisions and operations research*, doi: 10.22105/dmor.2021.267139.1302. (In Persian)