

Multi-Objective Optimal Model for Task Scheduling and Allocation in a Two Machines Robotic Cell Considering Breakdowns

BAHAREH VAISI¹, HIWA FARUGHI², SADIGH RAISSI³

Young Researchers and Elite Club¹, Department of Engineering², School of Industrial Engineering³
South Tehran Branch Islamic Azad University^{1,3}, University of Kurdistan²
Tehran^{1,3}, Sanandaj²

IRAN

st_b_vaisi@azad.ac.ir h.farughi@uok.ac.ir Raissi@azad.ac.ir

Abstract: - This paper aimed to demonstrate a metaheuristic as a solution procedure to schedule a two-machine, identical parts robotic cell under breakdown. The proposed previous model enabled one to determine optimal allocation of operations to the machines and corresponding processing times of each machine. For the proposed mathematical model to minimize cycle time and operational cost, multi-objective particle swarm optimization (MOPSO) algorithm was provided. Through some numerical examples, the optimal solutions were compared with the previous results. MOPSO algorithm could find the solutions for problems embeds up to 50 operations in a rationale time.

Key-Words: - Robotic manufacturing cell, Scheduling, Breakdowns, multi-objective particle swarm optimization algorithm.

Received: July 26, 2019 . Revised: December 16, 2019. Accepted: January 15, 2020. Published: January 31, 2020.

1 Introduction

Almost all flexible manufacturing systems are comprised of robots, CNC machines and other relevant stand-alone systems such as inspection machines, instrumentation devices, computers and sensors. The responsibility of robot is to pick-up products, load/unload machines and also material transposition inside the robotic cell.

To improve the system's productivity, there are several researches about the sequencing of machine feedings and robot movements in robotic cells. Machine breakdowns and transportation times so far have been relaxed while they may result in changing the solutions.

Although in most of the previous researches conducted in the field of robotic manufacturing cells such as [28], [29], [17], [30], [7], [15], [6], [16], [10], [8] and [3], scheduling is done based on a single criterion, the most important results of multi criteria scheduling were surveyed in [14]. Bi-criteria scheduling model in a two-machine robotic cell which produces identical parts was presented by Gultekin et al. [11]. They assumed that, the allocations of operations and their processing times are decision variables. Bi-objective mixed integer programming scheduling model in a cyclic robotic cell with processing time windows and non-Euclidean travel times was developed in [9]. ϵ -constraint method was proposed to solve the model. For studying other bi objective robotic cell scheduling papers, the reader is referred to [21], [2], [18], [12], [4], [33] and [24].

In a new paper a dynamic scheduling problem was addressed by Ma et al. [23]. In their robotic cell, more than one new jobs were arrived and should be scheduled immediately. The problem was formulated as a mixed integer programming model then a hybrid algorithm was proposed to search for a near-optimal solution.

One of the basic assumptions in aforementioned studies was that, there isn't any need for maintenance because robots and machines never experience failure. Moreover, in none of the previous studies, availability of robotic cell was recommended as a constraint, because in all of them machine/robot were assumed to be available. But in real world it is impossible, so to make the robotic cell scheduling issue more practical, no failure assumption for the machines was relaxed (see papers [26], [27] and [13]).

In this study, we developed a stochastic model for an unreliable robotic cell under different operational conditions comprising failures and preventive maintenance. The system was served by a single gripper robot for load/load identical parts as well displacements. Considering condition-based maintenance and its impact on the processing time of operations in a two-machine robotic cell, the focus of study is on S2 as the most commonly used robot's move cycle. Additionally, robotic cell's availability was considered as a constraint. As availability improvement will increase the output of the robotic cell, making an appropriate balance between cycle time and total operational cost,

considering breakdowns, are our objective in this study.

This paper is presented as follows: in section 2 the problem definition is demonstrated. In section 3 MOPSO as a solution procedure for the S_2 cycle is given and through numerical examples sensitivity analysis about results and discussion based on the previous proposed model is revealed in section 4. Finally, conclusion is presented in section 5.

2 Problem Statement

Decreasing production cost and rising quality is manageable by applying a flexible system. Such systems, consist of one or more machines, supported by a robot for load/load of parts. A typical in-line two-machine robotic cell based on [11] is shown in Fig.1.

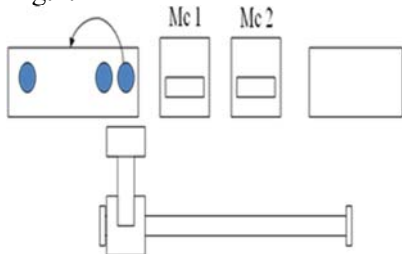


Fig.1. A typical layout for a 2-machine in-line robotic cell

In the production system there are two identical CNC machines that each of which has no priority in operation. Typically, in a 2-machine cell, three cycles, S_1 , S_2 and $S_{12}S_{21}$, is being applied to part displacements. As mentioned before, we focused on the S_2 cycle because it is a commonly used cycle with more complexity rather than the others. As a well-known rule, the activity sequence of S_2 cycle is coded by $\langle 01 A_{23} A_{12} \rangle$ [1]. Where A_{pq} denotes the robot activity sequence from station, p , to station q .

The literature revealed that, scheduling of flexible manufacturing cells commonly was carried out in deterministic conditions and little research has done on this issue under uncertainty. None of published papers has focused on scheduling-allocation optimization in the presence of random failures. For the scheduling-allocation optimization, the assumptions, parameters and proposed mathematical model are equivalent to [34]. In this study, we want to examine another solution method and compare the results.

3 Solution Methodology

In this paper, we applied MOPSO algorithm to generate different sets of non-dominated solutions

for the model. Then, results were compared with the results of ϵ -constraint approach in [34].

3.1 MOPSO

The simplicity, low computation cost and increasing popularity of Multi-Objective Particle Swarm Optimization, enhance its efficiency to solve simple as well as complex problems [22]. Before describing MOPSO algorithm, the particle swarm optimization (PSO) needs to be presented, shortly.

PSO is a computational method to optimize a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO is originally attributed to Kennedy, Eberhart and Shi and was first intended for simulating social behavior as a stylized representation of the organisms movements in a bird flock or fish school [19], [31], [20]. The analogy of PSO with evolutionary algorithms makes evident the notion that using a Pareto ranking scheme could be the straightforward way to extend the approach to handle multi objective optimization problems [5]. In PSO, first, some predefined particles are generated incidentally in the solution space. The situation of each particle represents an order of alternatives. Besides, the fitness function value for algorithm is given identically as the determined score for the considered problem. In each iteration, each particle should be moved based on other particles situations. Two types of solutions have to be updated in each iteration of this algorithm. The first variable which is denoted by $PBest$ is the best permutation experienced by each particle while the second one named as $GBest$ is the best experienced arrangement by all particles. Particles movement direction and their final position in each iteration will be calculated according to the following equations [32]:

$$V_i(t+1) = w \cdot V_i(t) + c_1 r_1 (PBest_i(t) - X_i(t)) + c_2 r_2 (GBest(t) - X_i(t)) \quad (19)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (20)$$

To control the impact of the previous history of velocities on the current velocity of a given particle, w (Inertia Weight) is employed. c_1 and c_2 are learning factors and are usually defined as constants. In order to apply the PSO strategy for solving multi-objective optimization problems, it is obvious that the original scheme must be modified. Clearly, the solution set of a problem with multiple objectives does not consist of a single solution (as in global optimization).

Instead, in multi-objective optimization, we aim to find a set of different solutions (the so-called Pareto optimal set) [25].

4 Test Problems, parameter Tuning and Software Implementation

A set of problems with different sizes from [34] are considered to test the performance of MOPSO method in comparison with ϵ -Constraint in this section.

Example. Let us consider three different Test Problems including different groups of operations with their processing times in a 2-machine robotic cell for producing identical parts. Values are given in Table 2. The main difference between these test problems is the number of operations (i.e. group size). The parameters and user defined values for the considered robotic cell are presented in Table 1. It should be noted that, the same tool is used for all of these operations and we assumed these parameter values are constant.

Table1. Characteristics of required parameters

$C_{PM}=35$	$B=80$	$\mu=2$
$C_o=50$	$k=4$	$\lambda=3$
$C_{TOOL}=45$	$C=7$	$\epsilon=1$
$A=40$	$H=15$	$\delta=2$
$t_{j,PM}=7$	$t_{j,CM}=10$	

4.1 MOPSO Results

To express the performance of the proposed model to solve the problems, we test the proposed MOPSO on a set of instances in Table 2 with MATLAB software and show the results in Table 4. It should be noted that, MOPSO algorithm was run on a portable PC with MS-Windows Vista, 3.0 GB of RAM, and 2.0 GHz Core 2 Duo CPU. Parameter setting for MOPSO algorithm was based on Table 3 and the algorithm procedure is shown in Fig.2. In Fig.3, we represent the best results of feasible region for objective functions in Test Problems based on MOPSO algorithm. Similar to the previous paper [34], as an instance, regarding the machine's generated degradation level, we demonstrate the allocation of operations to them for the Example Number 9 (see Table 2) in Table 5.

Table3. The MOPSO parameter setting

Population size	50
Repository Size	50
Personal Learning Coefficient	1
Global Learning Coefficient	2
Grid Inflation	0.1
Number of Grids	5
Leader Selection Pressure	2
Repository Member Selection Pressure	2
Maximum Number of Iterations	100

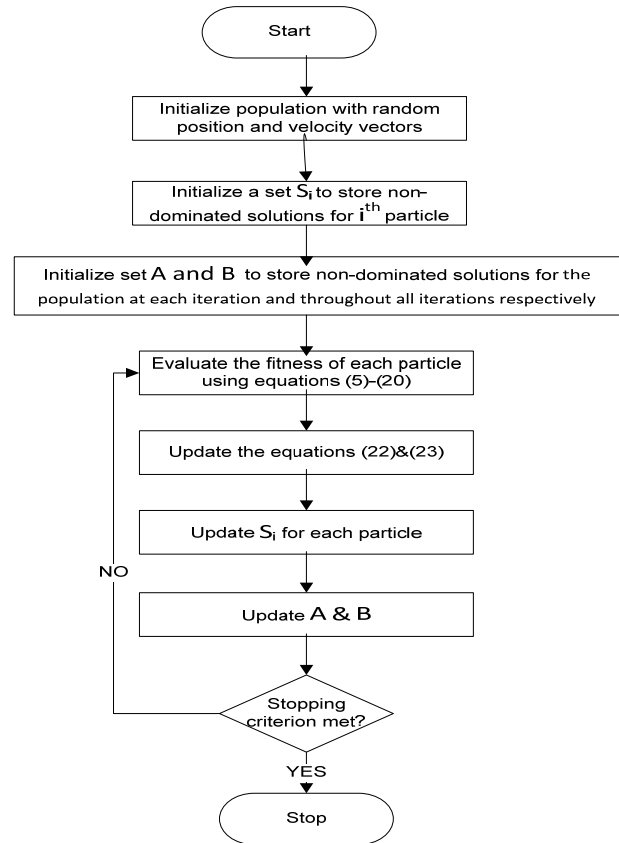


Fig.2. Multi-Objective Particle Swarm Optimization flowchart

Table4. The MOPSO results (lower bounds) through designated test problems

Example #	Min cost	Min S ₂ Cycle time	Elapsed time in seconds
1	1655.47	22	20.19
2	2555.31	27	23.49
3	2702.28	29	22.04
4	3252.61	34	20.72
5	4200.71	44	23.23
6	4851.59	50	21.26
7	5201.86	54	19.81
8	5551.34	57	25.11
9	5700.28	59	25.19
10	8049.49	82	24.86
11	11350.03	115	27.03
12	15648.31	158	31.08

Table 5. Allocation of operations for case 9 in the test problems based on MOPSO algorithm

Machine #	Allocated operations #
1	1, 5, 8, 10, 12, 13, 14, 18
2	2, 3, 4, 6, 7, 9, 11, 15, 16, 17

4.1.1 Comparison Metrics

We used diversity and spacing metrics to provide a basis for assessing the relative performance of MOPSO as a multi-objective optimization algorithm. The metrics definition is being summarized in Table 6:

Table 6. The comparison metrics

Spacing metric: (SM)	$d^- = \frac{\sum_{i=1}^n d_i}{n}$ $SP = \sqrt{\frac{\sum_{i=1}^n (d^- - d_i)^2}{n - 1}}$
Diversity metric: (DM)	$DM = \sqrt{\sum_{i=1}^n \max(\ x_i - y_i\)}$

Where $\|x_i - y_i\|$ is Euclidean distance between two non-dominated solutions x_i and y_i [35]. Comparison metrics were calculated for all test problems (see Table 2) over 10 runs of MOPSO algorithm. Table 7 is presented the diversity and spacing metrics obtained for small, large and big size instances.

4.2 Statistical analysis

To compare the outcome of both Pareto optimal solution methods (ϵ -constraint and MOPSO), we define objective functions result and elapsed time required by each method as the criteria. The criteria values have been summarized in Table 8.

4.2.1 Comparison between Solution Methods

To compare the performance of ϵ -constraint method with multi-objective particle swarm optimization algorithm, firstly, ϵ -constraint method's minimum cost was measured against that of MOPSO algorithm. The first objective function (i.e. cost) in exact solution is consistently less than meta-heuristic solution, regarding Fig.4. In considered Test Problems (see Table 2), the difference between objective function values is at least 15 and at most 122 currencies.

5 Conclusion

The studied system consists of a two-machine robotic manufacturing cell which produces identical

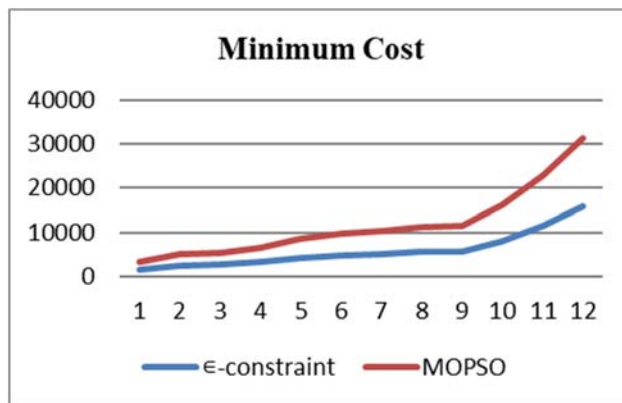


Fig.4. Solution procedures comparison of the first objective function in designated Examples

In the second objective function (i.e. S_2 cycle time) a statistical significant difference between these two solution methods is not observed and precisely describing, the test of equality between these solution procedures based upon Test Problems result, is significant at 99.7% significance level with 190 as non-parametric Mann-Whitney statistics and p-value of 0.4728, by applying Minitab™. Consequently, there is no difference between the two solutions method behavior in terms of S_2 cycle time.

From the standpoint of elapsed time, the average elapsed time for running MOPSO algorithm is repeatedly lower than that of ϵ -constraint method, concerning Fig. 5. In the considered Test Problems (see Table 2), the smallest difference between elapsed time amounts is 1 and the maximum difference between elapsed time values is 3569 Seconds.

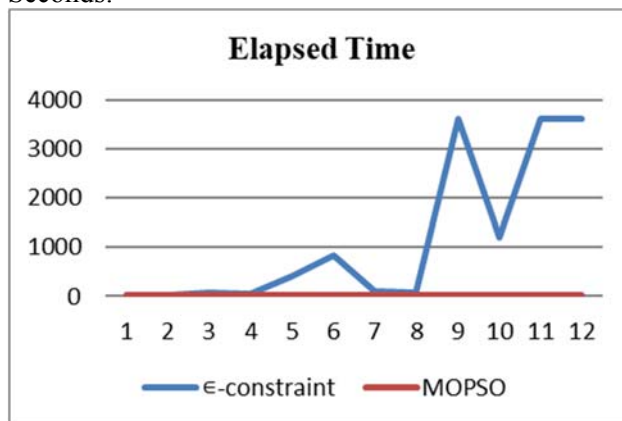


Fig.5. Solution procedures comparison of the elapsed time in designated Examples

parts and the robot moves cyclic on the basis of S_2 cycle. This robotic cell faces failures and repairs. This study aimed to demonstrate other solution procedure for the previous proposed model for the

defined above system and a generated set of Pareto optimal solutions based on MOPSO was presented. The link between operating conditions decisions in robotic manufacturing cells and maintenance decisions through following up maintenance task,

will improve time and operational costs simultaneously.

We believe that, the model and solution procedures could be extended to the robotic cell considering robot breakdowns or to the dual-gripper robot.

Table2. Test problems

Test problem	Group size	Example #	Processing times
1 (Small size)	(5-10) operations	1	10, 8, 7, 4, 3.
		2	10, 7, 13, 8, 5, 2, 5.
		3	7, 4, 3, 7, 4, 8, 10, 3, 7.
2 (Large size)	(10-18) operations	4	10, 8, 7, 4, 3, 7, 4, 8, 10, 3.
		5	10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 7, 4, 8.
		6	10, 7, 13, 8, 5, 2, 5, 4, 10, 7, 4, 8, 10, 3.
		7	10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7.
		8	10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 5, 2.
3 (Big size)	(18-50) operations	9	10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 5, 2, 3.
		10	10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 10, 8, 7, 4, 3.
		11	10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 10, 8, 7, 4, 3, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 2.
		12	10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 10, 8, 7, 4, 3, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 2, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 1, 2, 3, 7, 9.

Table7. Computational results for comparison metrics in test problems

Test Problem	Example #	DM	SM	Test Problem	Example #	DM	SM
Small size	1	153.0380	0	Large size	7	270.2473	0
	2	190.4197	0		8	279.2708	0
	3	195.6059	0	Big size	9	283.0309	0
Large size	4	214.2844	0		10	335.6310	0
	5	243.2365	0		11	398.7655	0
	6	260.9949	0		12	467.9246	0

Table8. Summary of the objective functions result and elapsed time for Pareto optimal solutions

Example #	Criteria					
	Min cost		Min S2 Cycle time		Elapsed time (in seconds)	
	€- constraint	MOPSO	€- constraint	MOPSO	€- constraint	MOPSO
1	1683.457	1655.4687	23	22	22	20.185975
2	2601.222	2555.3125	30	27	18	23.488227
3	2720.021	2702.2802	35	29	68	22.036794
4	3267.914	3252.6064	44	34	47	20.715749
5	4259.988	4200.7082	49	44	409	23.224983
6	4919.679	4851.5939	60	50	824	21.258663
7	5223.75	5201.8659	60	54	114	19.805676
8	5623.888	5551.3398	59	57	69	25.112806
9	5801.277	5700.2773	59	59	3600	25.191318
10	8131.095	8049.487	84	82	1211	24.857310
11	11373.37	11350.0269	122	115	3600	27.025602
12	15770.43	15648.3065	158	158	3600	31.076154

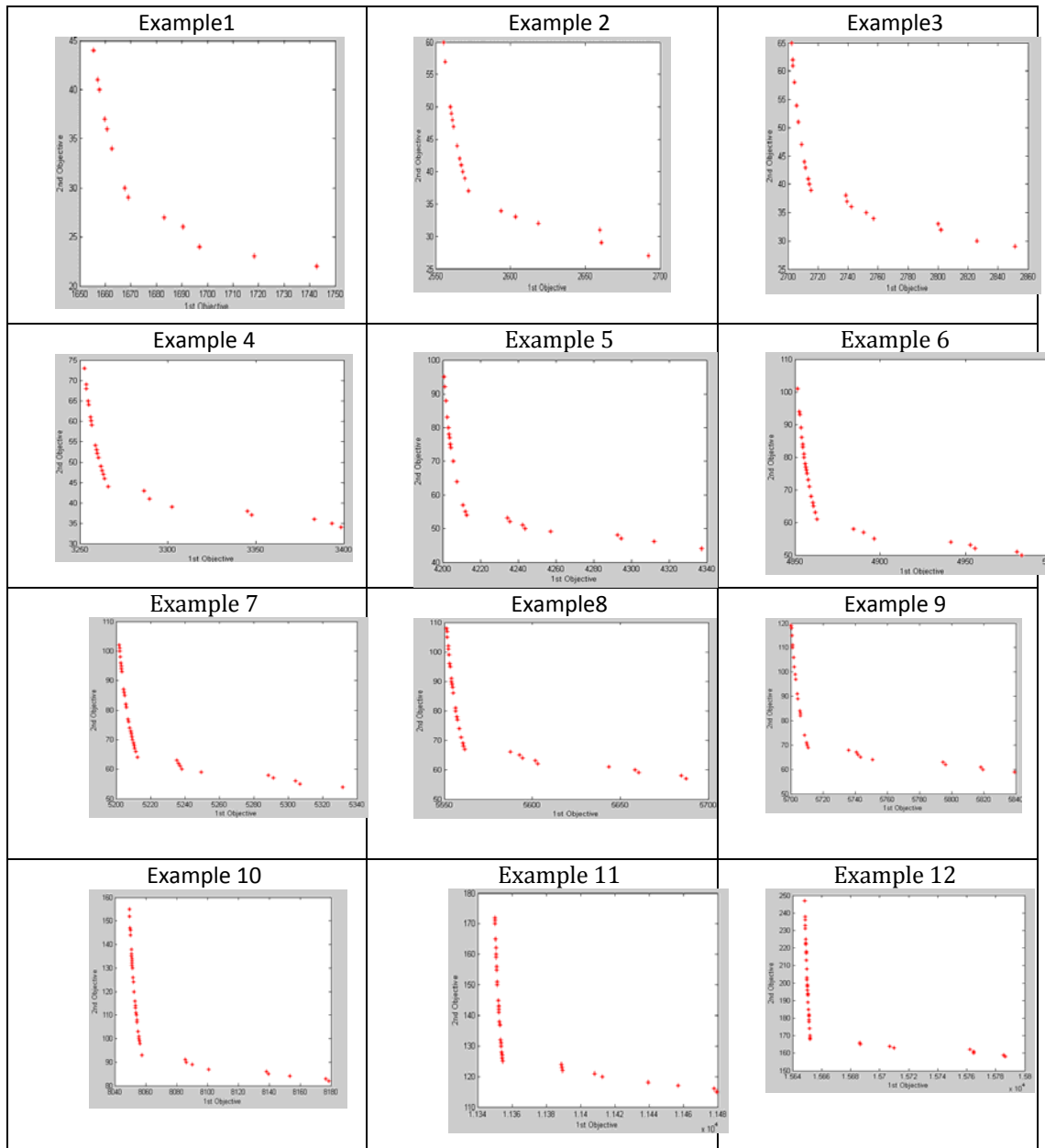


Fig.3. Results of feasible region for objective functions in designated Examples based on MOPSO algorithm

References:

- [1] Akturk, M.S., Gultekin, H., Karasan, O.E., "Robotic Cell Scheduling with Operational Flexibility", *Discrete Applied Mathematics*, Vol. 145, 2005, pp. 334–348.
- [2] Alagoz, O. and Azizoglu, M. (2003) "Rescheduling of Identical Parallel Machines under Machine Eligibility Constraint", *European Journal of Operational Research*, 149: 523–532.
- [3] Al-Salem, M., Haouari, M., Kharbeche, M., & Khallouli, W. (2016). A Free-Slack-Based Genetic Algorithm for the Robotic Cell Problem with Controllable Processing Times. In *Heuristics, Metaheuristics and Approximate Methods in Planning and Scheduling* (pp. 77-93). Springer International Publishing.
- [4] Brintrup, A.M. and Ramsden, J., Tiwari, A. (2007) "An Interactive Genetic Algorithm based Framework for Handling Qualitative Criteria in Design Optimization", *Computers in Industry*, 58 (3): 279-291.
- [5] Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3), 256-279.
- [6] Dawande, M., Geismar, H., Sethi, S. P., Sriskandarajah, C. (2005). "Sequencing and Scheduling in Robotic Cells:Recent Developments", *Journal of Scheduling*, 8(5): 387-426.
- [7] Dawande, M., Sriskandarajah, C. & Sethi, S. P. (2002) "On Throughput Maximization in Constant Travel-time Robotic Cells", *Manufacturing and Service Operations Management*, 4:296-312.
- [8] 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).
- [9] 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.
- [10] Geismar, H. N., & Pinedo, M. (2010). Robotic cells with stochastic processing times. *IIE Transactions*, 42(12), 897-914.
- [11] Gultekin, H., Akturk, M. S., Karasan, O.E. (2010) "Bi Criteria Robotic Operation Allocation in a Flexible Manufacturing Cell", *Computers & operations research*, 48(4): 779-789.
- [12] Gurel, S. and Akturk, M. S. (2007). "Considering Manufacturing Cost and Scheduling Performance on a CNC Turning Machine", *European journal of operational research*. 177(1):325-343.
- [13] Hamasha, M. M., Alazzam, A., Hamasha, S., Aqlan, F., Almeanazel, O., Khasawneh, M. T. (2015). "Multi machine Flexible Manufacturing Cell Analysis Using a Markov Chain-Based Approach", *Components, Packaging and Manufacturing Technology*, *IEEE Transactions on*, 5(3): 439-446.
- [14] Hoogeveen, H. (2005) "Multicriteria Scheduling", *European Journal of Operational Research*, 167: 592–623.
- [15] Hurink, J. and Knust. S.A. (2002) "Tabu Search Algorithm for Scheduling a Single Robot in a Job-Shop Environment", *Discrete Applied Mathematics*, 119:181–203.
- [16] Kamal Abadi, I.N., Gholami, S. (2009). "Robot Movements in a Cyclic Multiple-Part Type Three-Machine Flexible Robotic Cell Problem", *Transaction E: Industrial Engineering c Sharif University of Technology*, 16(1): 36-54.
- [17] Kamoun, H., Hall, N. G., SRISKANDARAJAH, C. (1999) "Scheduling in Robotic Cells: Heuristics and Cell Design", *Operations Research*, 47(6): 821-835.
- [18] Kayan, R.K. and Akturk, M.S. (2005) "A New Bounding Mechanism for the CNC Machine Scheduling Problems with Controllable Processing Times", *European Journal of Operational Research*, 167: 624–643.
- [19] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks. IV.* pp. 1942–1948.
- [20] Kennedy, J. (1997, April). The particle swarm: social adaptation of knowledge. In *Evolutionary Computation, 1997.*, *IEEE International Conference on* (pp. 303-308). IEEE.
- [21] Koksalan, M., Azizoglu, M., Kondakci, S. K. (1998) "Minimizing Flow Time and Maximum Earliness on a Single Machine", *IIE TRANSACTIONS*, 30(2): 192-200.
- [22] Lalwani, S., Singhal, S., Kumar, R., & Gupta, N. (2013). A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm. *Transactions on Combinatorics*, 2(1), 39-101.

- [23] Ma, K., Yan, P., & Dai, W. (2016, June). A hybrid discrete differential evolution algorithm for dynamic scheduling in robotic cells. In *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on* (pp. 1-6). IEEE.
- [24] Restrepo, I. M., Balakrishnan, S. (2008) "Fuzzy-based Methodology for Multi-Objective Scheduling in a Robot-Centered Flexible Manufacturing Cell", *J Intell Manuf*, 19: 421-432.
- [25] Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3), 287-308.
- [26] Savsar, M., Aldaihani M. (2008), "Modeling of machine failures in a flexible manufacturing cell with two machines served by a robot", *Reliability Engineering & System Safety*, 93 (10): 1551-1562.
- [27] Savsar, M. (2010), "Reliability modeling of a manufacturing cell operated under degraded mode", In *Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management*, Dhaka, Bangladesh.
- [28] Sethi, S. P. and d. Groupe d'études et de recherche en analyse des (1989) "Sequencing of Robot Moves and Multiple Parts in a Robotic Cell", *Montréal Groupe d'études et de recherche en analyse des décisions*.
- [29] 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: 331-358.
- [30] Sethi, S., Sriskandarajah, C. and Sidney, J. (2001) "Scheduling in Dual Gripper Robotic Cells for Productivity Gains", *IEEE Transactions on Robotics and Automation*, 17: 324-341.
- [31] Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, The 1998 IEEE International Conference on (pp. 69-73). IEEE.
- [32] Toroudi, H., Madani, M., Sarlak, F., & Kanani, Y. (2017). A multi-objective method for solving assembly line balancing problem. *Decision Science Letters*, 6(1), 1-10.
- [33] Turkcan, A., Akturk, M. and Storer, R.H. (2007) "Due Date and Cost-based FMS Loading, Scheduling and Tool Management", *International Journal of Production Research*, 45(5): 1183-1213.
- [34] 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.
- [35] Zhu, Q., Peng, H., & van Houtum, G. J. (2015). A condition-based maintenance policy for multi-component systems with a high maintenance setup cost. *OR Spectrum*, 37(4): 1007-1035.