Dynamic Distributed Job-Shop Scheduling Problem Consisting of Reconfigurable Machine Tools

Mehdi Mahmoodjanloo, Reza Tavakkoli-Moghaddam, Armand Baboli, Ali Bozorgi-Amiri

HAL Id: hal-03635652
https://inria.hal.science/hal-03635652

Submitted on 20 Jun 2023

Dynamic Distributed Job-Shop Scheduling Problem Consisting of Reconfigurable Machine Tools

Mehdi Mahmoodjanloo[1,(✉)], Reza Tavakkoli-Moghaddam[1] Armand Baboli[2],
Ali Bozorgi-Amiri[1]

[1] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[2] LIRIS laboratory, UMR 5205 CNRS, INSA of Lyon, 69621 Villeurbanne cedex, France
mehdi.janloo@ut.ac.ir

Abstract. Keeping pace with rapidly changing customer requirements forces companies to increase the capability of adaptation of their production systems. To fulfill the market requirements in a reasonable time and cost, distributed manufacturing has been emerged as one of the efficient approaches. Moreover, the ability of reconfigurability makes manufacturing systems and tools to be more adaptable. This research deals with a dynamic production scheduling problem simultaneously in several different shop-floors consisting of reconfigurable machine tools (RMTs) by utilizing the real-time data extracted from a cyber-physical system (CPS). First, a mathematical programming model is presented for the static state. Thereafter, by utilizing the CPS capabilities, a dynamic model is extended to schedule new jobs, in which there have already been some other jobs in each facility. A numerical example is solved to illustrate the validation of the model. Finally, some potential solving approaches are proposed to make the model implementable in real-world applications.

Keywords: Distributed manufacturing, Reconfigurable manufacturing systems, Dynamic scheduling, Industry 4.0, Cyber-physical shop floor.

1   Introduction

Over the last decades, due to the globalization of the economy and its requirements, enterprises aim to reach higher operational performance at a lower cost. Hence, distributed manufacturing has been emerged as one of the efficient approaches to overcome these requirements. On the other hand, capabilities of new emerging smart factories are enhancing supported by Industry 4.0 technologies, such as a cyber-physical system (CPS), cloud computing, virtual reality, internet of things (IoT) and big data. To gain competitive advantages of these technologies, companies are going to adapt for a more customer-oriented market [1]. Also, new manufacturing systems need to satisfy the main requirements of this market, such as the rapid introduction of new products and high demand fluctuations. A significant approach to cope with these issues is to take advantages of reconfigurability for manufacturing systems and tools. Hence, a new class of production machines, called reconfigurable machine tools (RMTs), have been introduced. An RMT machine usually has a modular structure, which makes it be able

to obtain different configurations in order to satisfy manufacturing requirements. One of the benefits of developing RMTs is that the use of several different machines that share many costly and common modules while being rarely used at the same time can be prevented [2]. The capacity and functionality of a manufacturing system by relying on abilities of such machines can be adjusted easily. Despite the special capabilities, it seems that the studies in production scheduling in such systems face new challenges considering the high level of dynamism and complexity of needed decisions.

De Giovanni and Pezzella [3] studied a static distributed and flexible job-shop scheduling problem. They developed an improved genetic algorithm (GA) to solve the problem efficiently. However, they did not consider a dynamic environment of the production system. Tian et al. [4] designed a production information management system based on industrial IoT technology to tackle the dynamic flexible job-shop scheduling problem in a rolling horizon. Romero-Silva and Hernández [5] studied in different manufacturing contexts the role of the CPS to provide a company by carrying out a better scheduling task. They found that production systems with uncertain demands and complex production processes could benefit the most from implementing a CPS at a shop-floor level. However, they did not consider the effect of reconfigurability on the studied systems. Recently, Mahmoodjanloo et al. [6] presented a new variant of a job-shop scheduling problem, which contains reconfigurable machine tools. They developed two static models to tackle the scheduling problem in a centralized production system.

In this paper, by utilizing the cyber-physical shop-floor capabilities, a distributed flexible job-shop scheduling problem consisting of reconfigurable machine tools in a dynamic environment is studied.

2   Static and Dynamic Model Presentation

2.1   Problem description

There is a set $\mathcal{J}$ of n jobs, where each one should be processed in one of the existing facilities. Job i has a set of n_i operations with a predefined sequence (e.g., $O_{i,1} \rightarrow O_{i,2} \rightarrow \cdots \rightarrow O_{i,n_i}$). The facilities have already been deployed in different geographical areas. In each facility $f \in F$, there is a set of RMTs, in which each RMT $k \in K_f$ has a set of $C_{k,f}$ configurations. It is assumed that operation O_{ij} can be processed at least on one configuration of one of the existing RMTs in each facility. No setup is needed to perform operations in a machine configuration, while to switch to a different configuration on the machine, the RMT needs to a setup that is dependent on two consecutive configurations. The main decisions of the problem include assigning of each job to one facility, allocating of each job operation to an eligible RMT, sequencing of the jobs and determining the appropriate configurations of each machine to perform the allocated operations. The objective is to minimize the total cost.

The scheduling environment can be classified into two main classes; namely, static (offline) and dynamic (online) scheduling. We first present the static model formulation, in which there are several jobs to be processed in the empty facilities. Thereafter, the model will be extended for the dynamic state, in which there are several jobs in each

facility that are processing based on a current schedule. By arriving some new jobs, the current schedules should be updated based on a variable-order rescheduling strategy, in which all unprocessed operations can be rescheduled after assigning the new arriving jobs to the facilities.

2.2 Static model formulation

Sets and indices:

$\mathcal{J}$ Set of jobs, where job index $i \in \mathcal{J}$

N_i Set of operations related to job i, where operation j of job i is denoted by O_{ij}

F Set of facilities, where facility index $f \in F$

K_f Set of machines, where machine index $k \in K_f$

$C_{k,f}$ Set of configurations of machine k in facility f, where configuration index $c \in C_{k,f}$

$L_{k,f}$ Set of job positions on machine k of facility f, where position index $l \in L_{k,f}$ (i.e., $\left| L_{k,f} \right| = \sum_{i \in J} \sum_{j \in N_i} \left(\max_{c \in C_{k,f}} R_{ijkc}^f \right)$ and $L = \bigcup_{f \in F, k \in K_f} L_{k,f}$).

Parameters:

PT_{ijkc}^f Processing time of operation O_{ij} on configuration c of machine k in facility f

R_{ijkc}^f Binary parameter. If operation O_{ij} can be processed on configuration c of machine k in facility f, then $R_{ijkc}^f = 1$; otherwise, $R_{ijkc}^f = 0$.

PC_{ijkc}^f Processing cost of operation O_{ij} on configuration c of machine k in facility f

TT_i^f Transportation time of job i from facility f

TC_i^f Transportation cost of job i from facility f

$ST_{c_1,c_2,k}^f$ Configuration-dependent setup time when configuration is changed from c_1 to c_2 (i.e., $c_1 \neq c_2$ and $c_1, c_2 \in C_{k,f}$) on machine k in the facility f

$SC_{c_1,c_2,k}^f$ Configuration-dependent setup cost when configuration is changed from c_1 to c_2 (i.e., $c_1 \neq c_2$ and $c_1, c_2 \in C_{k,f}$) on machine k in the facility f

D_i Due date of job i

ω_i Tardiness penalty of job i per time unit

Decision variables of the static model:

x_{fi} 1 if job i be assigned to facility f; 0, otherwise.

y_{ijklc}^f 1 if operation O_{ij} is processed on machine-positions k and l with configuration c of facility f; 0, otherwise.

$Z_{c_1,c_2,k,l}^f$ 1 if at the beginning of positions k and l of facility f, in which the machine's configuration is changed from c_1 to c_2 (i.e., $c_1 \neq c_2$); 0, otherwise.

CO_{ij} Completion time of operation O_{ij}

CP_{fkl} Finishing time of positions k and l in facility f

T_i Tardiness of job i

Mathematical formulation of the static problem:

$$\text{Min } z = \sum_{i \in \mathcal{J}} \sum_{j \in N_i} \sum_{f \in F} \sum_{k \in K_f} \sum_{c \in C_{k,f}} \sum_{l \in L_{k,f}} PC_{ijkc}^f \, y_{ijklc}^f + \sum_{i \in \mathcal{J}} \sum_{f \in F} TC_i^f \, x_{fi} + \sum_{i \in \mathcal{J}} \omega_i \, T_i \tag{1}$$
$$+ \sum_{f \in F} \sum_{k \in K_f} \sum_{c_1 \in C_{k,f}} \sum_{\substack{c_2 \in C_{k,f} \\ c_1 \neq c_1}} \sum_{l \in L_{k,f}} SC_{c_1,c_2,k}^f \, Z_{c_1,c_2,k,l}^f$$

Subject to:

$$\sum_{f \in F} x_{fi} = 1 \qquad \forall i \in \mathcal{J} \qquad (2)$$

$$\sum_{j \in N_i} \sum_{k \in K_f} \sum_{c \in C_{k,f}} \sum_{l \in L_{k,f}} y_{ijklc}^f \le n_i \times x_{fi} \qquad \forall f \in F, i \in \mathcal{J} \qquad (3)$$

$$\sum_{f \in F} \sum_{k \in K_f} \sum_{c \in C_{k,f}} \sum_{l \in L_{k,f}} y_{ijklc}^f = 1 \qquad \forall i \in \mathcal{J}, j \in N_i \qquad (4)$$

$$\sum_{l \in L_{k,f}} y_{ijklc}^f \le R_{ijkc}^f \qquad \forall i \in \mathcal{J}, j \in N_i, f \in F, k \in K_f, c \in C_{k,f} \qquad (5)$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in N_i} \sum_{c \in C_{k,f}} y_{ijklc}^f \le 1 \qquad \forall f \in F, k \in K_f, l \in L_{k,f} \qquad (6)$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in N_i} \sum_{c \in C_{k,f}} y_{ijk,l-1,c}^f \ge \sum_{i \in \mathcal{J}} \sum_{j \in N_i} \sum_{c \in C_{k,f}} y_{ijklc}^f \qquad \forall f \in F, k \in K_f, l \in L_{k,f}, l \ne 1 \qquad (7)$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in N_i} y_{ijk,l-1,c_1}^f + \sum_{i \in \mathcal{J}} \sum_{j \in N_i} y_{ijklc_2}^f \ge 2 Z_{c_1,c_2,k,l}^f \qquad \begin{matrix} \forall f \in F, k \in K_f, l \in L_{k,f}, \\ c_1, c_2 \in C_{k,f}, c_1 \ne c_2 \end{matrix} \qquad (8)$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in N_i} y_{ijk,l-1,c_1}^f + \sum_{i \in \mathcal{J}} \sum_{j \in N_i} y_{ijklc_2}^f - 1 \le Z_{c_1,c_2,k,l}^f \qquad \begin{matrix} \forall f \in F, k \in K_f, l \in L_{k,f}, \\ c_1, c_2 \in C_{k,f}, c_1 \ne c_2 \end{matrix} \qquad (9)$$

$$CO_{ij} \ge CO_{i,j-1} + \sum_{f \in F} \sum_{k \in K_f} \sum_{c \in C_{k,f}} \sum_{l \in L_{k,f}} PT_{ijkc}^f y_{ijklc}^f \qquad \forall i \in \mathcal{J}, j \in N_i \qquad (10)$$

$$CP_{fkl} \ge CP_{fk,l-1} + \sum_{c_1 \in C_{k,f}} \sum_{\substack{c_2 \in C_{k,f} \\ c_1 \ne c_2}} ST_{c_1,c_2,k}^f Z_{c_1,c_2,k,l}^f \qquad \forall f \in F, k \in K_f, l \in L_{k,f} \qquad (11)$$

$$+ \sum_{i \in \mathcal{J}} \sum_{j \in N_i} \sum_{c \in C_{k,f}} PT_{ijkc}^f y_{ijklc}^f$$

$$CP_{fkl} \le CO_{ij} + M \left(1 - \sum_{c \in C_{k,f}} y_{ijklc}^f \right) \qquad \forall f \in F, k \in K_f, l \in L_{k,f}, i \in \mathcal{J}, j \in N_i \qquad (12)$$

$$CP_{fkl} \ge CO_{ij} - M \left(1 - \sum_{c \in C_{k,f}} y_{ijklc}^f \right) \qquad \forall f \in F, k \in K_f, l \in L_{k,f}, i \in \mathcal{J}, j \in N_i \qquad (13)$$

$$CO_{i,n_i} + \sum_{f \in F} TT_i^f x_{fi} - T_i \le D_i \qquad \forall i \in \mathcal{J} \qquad (14)$$

$$y_{ijklc}^f, Z_{c_1,c_2,k,l}^f, x_{fi} \in \{0,1\} \qquad \forall f \in F, k \in K_f, c \in C_{k,f}, l \in L_{k,f}, i \in \mathcal{J}, j \in N_i \qquad (15)$$

$$C_{ij}, C'_{fkl}, T_i \ge 0 \qquad \forall f \in F, k \in K_f, l \in L_{k,f}, i \in \mathcal{J}, j \in N_i \qquad (16)$$

Eq. (1) is the objective function, which calculates total system costs. Constraint set (2) ensures that each job should be assigned to one of the facilities. Then, Constraint set (3) guarantees all operations of each assigned job be processed on the machines of the associated facility, where $n_i = |N_i|$, the number of operations which should be done on the job i. Constraint set (4) mandates each operation to be assigned to one position of an existing machine configuration of the associated facility. Constraint set (5) prevents the assignment of the operation O_{ij} to any positions of the configuration of machines k and c of facility f if it is not allowed to be performed on the configuration of machine c of machine k ($R_{ijkc}^f = 0$). Constraint set (6) guarantees that in each machine position

of a facility, at most one operation can be processed. Constraint set (7) shows that each position can be allocated only when the previous position has already been assigned. For each machine, Constraint sets (8) and (9) guarantee that a proper setup should be done if two consecutive positions be assigned to different configurations. Constraint set (10) shows that each operation O_{ij} can be started only after finishing of the previous operation. Constraint set (11) ensures that the finishing time of each machine position should be greater than the completion time of the previous position plus an associated processing time of the position and a possible setup time. Constraints (12) and (13) guarantee that the completion time of each machine position be set with its associated job operation (M is a big positive number). Constraint set (14) calculates the amount of tardiness for each job. Constraints (15) and (16) define the decision variables.

2.3   Dynamic model formulation

Herein, we extend the presented model for a dynamic situation, in which some new arriving orders should be scheduled to be produced in the manufacturing system. Utilizing the Industry 4.0 technologies, the required real-time data from the distributed shop-floors (e.g., delayed jobs, current configurations and accessibility of machines) and limitations of the logistics system (i.e., availability of raw materials) can be provided by a cyber-physical shop-floors (CPSF). The provided data can be utilized in the scheduling module to update the current schedules of each shop-floor. The shop-floors under an IoT environment can include three basic levels. The conceptual framework of information flow to provide the autonomous analysis of system status and real-time response to dynamic events is presented in Fig. 1.
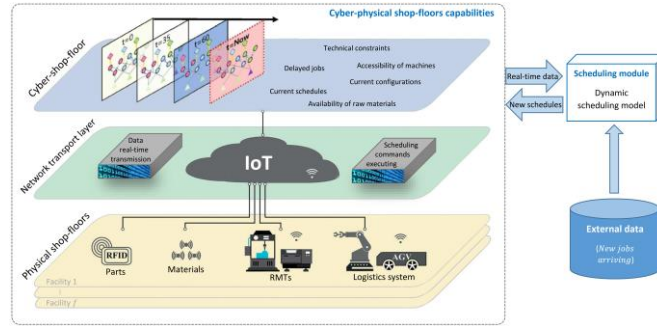
Fig. 1. Dynamic scheduling task utilizing CPSF capabilities

Updated sets and indices of the dynamic model:

$\mathcal{J}'$ Set of non-delivered jobs, where $\mathcal{J}' \subseteq \mathcal{J}$

$\mathcal{J}''$ Set of new arrived jobs

N_i' Set of all operations of new jobs (for $i \in \mathcal{J}''$) or remained operations of current jobs, where $N_i' \subseteq N_i$ for $i \in \mathcal{J}'$

$c_{k,f}'$ Index of current configuration of kth machine in facility f, where $c_{k,f}' \in C_{k,f}$

$L_{k,f}'$ Set of unoccupied machine-positions

$l'_{k,f}$ Index of the first unoccupied position of kth machine in facility f after arriving new jobs, where $L'_{k,f} = \{l'_{k,f}, l'_{k,f} + 1, \ldots, l'_{k,f} + |L'_{k,f}| - 1\}$

Considering the optimal value of the variables $x^*_{fi} \; \forall i \in \mathcal{J}$ and $CP^*_{f,k,l'_{k,f}-1}$ as the initial conditions of the dynamic model and substituting the sets $\mathcal{J}' \cup \mathcal{J}''$, N'_i and $L'_{k,f}$ respectively instead of the previous sets $\mathcal{J}$, N_i and $L_{k,f}$, the dynamic model can be defined as follows.

Mathematical formulation of the dynamic problem:

$$\text{Min } z = \sum_{i \in \mathcal{J}' \cup \mathcal{J}''} \sum_{j \in N'_i} \sum_{f \in F} \sum_{k \in K_f} \sum_{c \in C_{k,f}} \sum_{l \in L'_{k,f}} PC^f_{ijkc} \, y^f_{ijklc} + \sum_{i \in \mathcal{J}''} \sum_{f \in F} TC^f_i \, x_{fi} \qquad (17)$$

$$+ \sum_{i \in \mathcal{J}' \cup \mathcal{J}''} \omega_i \, T_i + \sum_{f \in F} \sum_{k \in K_f} \sum_{c_1 \in C_{k,f}} \sum_{\substack{c_2 \in C_{k,f} \\ c_1 \neq c_1}} \sum_{l \in L'_{k,f}} SC^f_{c_1,c_2,k} \, Z^f_{c_1,c_2,k,l}$$

Constraints $(4) - (13)$, (15) and (16) should be considered replacing updated the sets $\mathcal{J}' \cup \mathcal{J}''$, N'_i and $L'_{k,f}$ respectively instead of the previous sets $\mathcal{J}$, N_i and $L_{k,f}$.

$$\sum_{f \in F} x_{fi} = 1 \qquad\qquad\qquad\qquad \forall\, i \in \mathcal{J}'' \qquad\qquad (18)$$

$$\sum_{j \in N'_i} \sum_{k \in K_f} \sum_{c \in C_{k,f}} \sum_{l \in L'_{k,f}} y^f_{ijklc} \leq n_i \times x_{fi} \qquad \forall\, f \in F, i \in \mathcal{J}'' \qquad (19)$$

$$CO_{i,n_i} - T_i \leq D_i - \sum_{f \in F} TT^f_i \, x^*_{fi} \qquad \forall\, i \in \mathcal{J}' \qquad\qquad (20)$$

$$CO_{i,n_i} + \sum_{f \in F} TT^f_i \, x_{fi} - T_i \leq D_i \qquad \forall\, i \in \mathcal{J}'' \qquad\qquad (21)$$

3   Computational Results and Discussions

In this section, a small-sized example is illustrated to validate the proposed models. The main data of the example are presented in Table 1. At first, jobs 1 to 4 are considered to be scheduled by the static model. The optimum schedule is presented in Fig. 2. Thereafter, it is supposed that job 5 is arrived at $t = 85$. Moreover, based on the obtained real-time data from the CPSF, it is estimated that the needed time for providing the manufacturing system in facility 1 and 2 are 90 and 45 time units, respectively. In the optimum solution of the dynamic model, the new arriving job is assigned to the facility 1. As can be seen in Fig. 3, the unprocessed operations until $t = 175$ containing $O_{1,3}, O_{1,4}, O_{3,3}$ and all operations of job 5 are rescheduled.

A sensitive analysis on Δt shows that the total cost is strongly affected by this parameter. Hence, the performance of the manufacturing system directly is related to the response time. To make the method applicable in real-world problems, the system must be equipped with some major "hard" and "soft" technological requirements. A significant issue to improve the performance is to enhance the agility of the logistics system.

Developing an appropriate smart logistics system can be helpful to make the manufacturing system much more flexible. Also, the decision-making approach and the related computational time should be considered. For example, the proposed models is coded in GAMS 24.1.3 software and solved by CPLEX solver. The written codes are performed on a computer with a 2/5 GHz Intel Core(TM) i7-4710HQ CPU and 8 GB RAM. The static and dynamic models use 496 and 273 seconds of CPU time to be solved, respectively. Hence, developing an efficient algorithm to solve the problem in a reasonable time can be interesting.

Table 1. Data for the illustrated example

f,k,c	Job1				Job2					Job3			Job4				Job5		
	1	2	3	4	1	2	3	4	5	1	2	3	1	2	3	4	1	2	3
1, 1,1					75			51			87	91		52	57	72		65	74
1, 1, 2		72		86	54					96							53		
1, 2, 1		65	55	61		62	81		86			59	86		76	54			
1, 2, 2	93	97	63				90			67	96	62					76	50	62
2, 1, 1	81		71				70	55		83	85		95			83		71	
2, 1, 2				80		97	77		63	75	60								55
2, 1, 3	60	58					99							84				82	
2, 2, 1					75			51			87	91		52	57	72			
2, 2, 2		72		86	54					96							53		
D_i, w_i	430, 100				650, 150					400, 65			400, 50				650, 70		
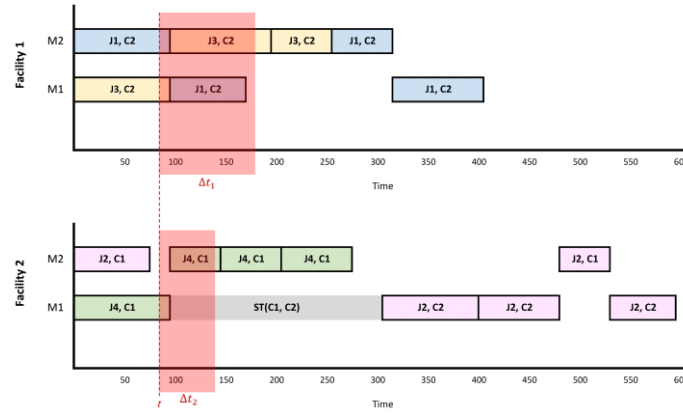
Fig. 2. Optimal schedule of the illustrated example with $z^* = 153500$
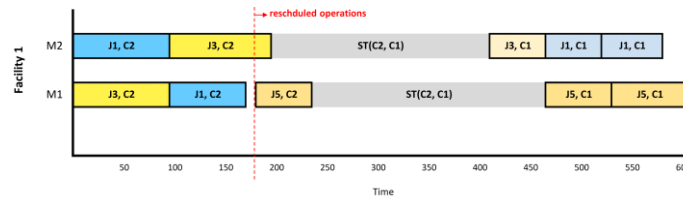
Fig. 3. Rescheduled operations of the illustrated example $z^*_{new} = 210110$

4   Conclusions

Because of current globalization trend, many manufacturing systems have shifted to a distributed production network instead of a single factory production. Moreover, information technology has been utilized to make producers more competitive in today's rapidly changing market requirements. These evolutions can affect many areas of decision making, including scheduling. In this paper, a mathematical model has been presented for a distributed flexible job-shop scheduling in a dynamic environment by utilizing cyber-physical shop-floor capabilities. The full adoption of the framework presented will help companies enhance their global competitiveness, improve the flexibility of their domestic manufacturing systems, and obtain new market opportunities.

Due to the different aspects of the decision environment, the high complexity of the problem, and a requirement for real-time decision making, a multi-agent approach can be effective to solve it. Moreover, because of high volume data generated in the shop-floor agents, machine learning approaches by doing a general analysis of data can be utilized to facilitate solving some of specific tasks.

References

1. Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019). "Review of job shop scheduling research and its new perspectives under Industry 4.0." *Journal of Intelligent Manufacturing*, **30**(4), 1809-1830.
2. Gadalla, M., & Xue, D. (2017). "Recent advances in research on reconfigurable machine tools: a literature review." *International Journal of Production Research*, **55**(5), 1440-1454.
3. De Giovanni, L., & Pezzella, F. (2010). "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem." *European journal of operational research*, **200**(2), 395-408.
4. Tian, S., Wang, T., Zhang, L., & Wu, X. (2019). "The Internet of Things enabled manufacturing enterprise information system design and shop floor dynamic scheduling optimisation." *Enterprise Information Systems*, 1-26.
5. Romero-Silva, R., & Hernández-López, G. (2019). "Shop-floor scheduling as a competitive advantage: A study on the relevance of cyber-physical systems in different manufacturing contexts." *International Journal of Production Economics*, **224**, 107555.
6. Mahmoodjanloo, M., Tavakkoli-Moghaddam, R., Baboli, A., & Bozorgi-Amiri, A. (2020). "Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm." *Applied Soft Computing*, **94**, 106416.