International Journal of Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281 Volume 3, Issue 5, May 2020

IJAMSR 3 (5) www.ijamsr.com CrossRef: https://doi.org/10.31426/ijamsr.2020.3.5.3311

IJAMSR

International Journal of
Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281

Job Shop Scheduling Considering The Assembly Process With NSGA-II

Zi-Yue Wang, Cong Lu

School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China

Email: conglu@uestc.edu.cn

Keywords:
Job shop scheduling
Assembly
NSGA-II
Optimization

A B S T R A C T

This paper proposes an approach for job-shop scheduling considering the assembly process (JSS-AP) in production, where job shop scheduling is carried out according to the given assembly sequence, with the objective to minimize the completion time and the inventory time during production. The mathematical models are established for job shop scheduling considering the assembly process, and the non-dominated sorted genetic algorithm-II (NSGA-II) is applied to solve the problem, by which the decision-making of the non-dominated solutions is carried out according to the decision maker's preference of the objectives in job shop scheduling. Based on the above, the framework of the proposed approach to solve job-shop scheduling considering the assembly process is given. Through the case study and comparison test, the proposed JSS-AP approach is verified to be able to shorten the lead time and save the manufacturing cost of the enterprise more effectively.

Citation: *Zi-Yue Wang, Cong Lu (2020). Job Shop Scheduling Considering The Assembly Process With NSGA-Ⅱ. International Journal of Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281, 3 (5), May 2020, Pp 1 - 11*

IJAMSR 3 (5) www.ijamsr.com CrossRef: https://doi.org/10.31426/ijamsr.2020.3.5.3311

International Journal of
Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281

Introduction

As the important technology in production planning, job shop scheduling can be applied to generate the optimal part processing plan, by which the manufacturing resources are assigned to different processing operations to shorten the makespan and save the manufacturing cost for the enterprise.

As a NP-hard and intractable combinatorial optimization problem, Job shop scheduling problem (JSSP) has received much research attention. Gong et al. [1] proposed an effective memetic algorithm to solve the multi-objective JSSP considering the makespan and tardiness criteria. Kurdi [2] proposed a hybrid island model genetic algorithm to solve the JSSP with the objective to minimize the makespan. Wang [3] proposed an adaptive job shop scheduling strategy considering the dynamic and uncertain production environment of job shops. Yang et al. [4] investigated job-shop scheduling through building a grey scheduling model, aiming to obtain the precise makespan or delivery period with a hybrid grey cuckoo search algorithm. Zhang et al. [5] proposed a hybrid discrete particle swarm optimization algorithm to solve the dual resource constrained JSSP with resource flexibility. Asadzadeh [6] proposed a parallel artificial bee colony algorithm to solve the JSSP, the communication between colonies was carried out by exchanging migrants with a dynamic migration strategy.

Besides above, Singh et al. [7] presented a particle swarm optimization algorithm for solving multi-objective flexible Job shop scheduling problem (FJSP) with the goal to optimize makespan, flow time, and tardiness. Zhu and Zhou [8] studied the FJSP with job precedence constraints, and an multi-objective grey wolf optimization algorithm was proposed to solve the problem with the optimization objectives including the makespan, machine workload and total machine workload. Xiong and Fu [9] developed an immune multi-agent scheduling system to solve the FJSP with the objective to minimize the maximal completion time. Caldeira and Gnanavelbabu [10] proposed an improved Jaya algorithm for solving the FJSP using the makespan criteria, where an efficient initialization mechanism, a local search technique and acceptance criterion were incorporated into the algorithm to improve the solution quality and maintain the diversity. Ajchara and Arit [11] developed a memetic algorithm based on marriage in honey bees optimization algorithm for solving FJSP. Dai et al. [12] investigated an energy-efficient FJSP with transportation constraints, an optimization model was formulated to optimize the objectives including the total energy consumption and the makespan.

In the above investigations of job shop scheduling problem, different algorithms were proposed to obtain the schedule which can minimize the objectives, such as the makespan, flow time. However, in these studies, the influence of the assembly process was not considered in job shop scheduling.

For the assembly job shop scheduling problem considering the influence of assembly operations, Fattahi et al. [13] proposed a method to solve flexible job shop scheduling problem with assembly operations, with the objective to minimize the completion time of all the products. Pereira and Santoro [14] developed an integrative scheduling method to simulate the operations scheduling process in assembly job shop scheduling systems. Allahverdi et al. [15] investigated the two-stage assembly flow shop scheduling problem with a bicriteria including the makespan and mean completion time, three heuristics algorithms were proposed to solve the problem. Lin et al. [16] presented a nonlinear integer programming model to formulate the flexible assembly job-shop scheduling problem with tight job constraints, with the goal to minimize the makespan of all products by arranging the operating sequence on each machine. Zhang et al. [17] studied the production scheduling problem in a flexible manufacturing system, where products were incorporated with flexible non-linear process plans and assembling operations, the makespan, total tardiness and total workload were taken as the optimization objectives. Liao et al. [18] investigated a two-stage assembly scheduling problem of N products with setup times, with the objective to minimize the makespan. Du et al. [19] proposed an assembly job shop scheduling model with the objective to minimize the total completion time, considering constraint conditions including process constraint, resources constraint, and assembly constraint.

From the above analysis, it can be found that the current researches have investigated the job shop scheduling problem considering the assembly operations, however the inventory time of parts during production was not considered in current job shop scheduling approaches considering the assembly process.

IJAMSR 3 (5) www.ijamsr.com CrossRef: https://doi.org/10.31426/ijamsr.2020.3.5.3311

International Journal of
Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281

The inventory time of parts is very important in the total production process especially for complex and large products such as the machine tool, where the part inventory not only takes up a large storage space, but also causes much inventory cost.

To address the above problem, this paper proposes an approach for job shop scheduling considering the assembly process (JSS-AP) in production, where job shop scheduling is carried out according to the given assembly sequence, with the objective to minimize the completion time and the inventory time during production.

2. Methods

To carry out job shop scheduling considering assembly process (JSS-AP) in production, the mathematical models are established in this section.

2.1. Notations

n: Number of parts $\{P_1, P_2, ..., P_n\}$

m: Number of processing machines $\{M_1, M_2, ..., M_m\}$

Q_i: Number of processing operations for part P_i

Sp_i: Starting time for processing part P_i

p_{ij}: The j-th operation of part P_i

Sp_{ij}: Starting time of the j-th operation of part P_i

Sp_{ijk}: Starting time of the j-th operation of part P_i on machine M_k

Tp_{ij}: Processing time of the j-th operation of part P_i

Tp_{ijk}: Processing time of the j-th operation of part P_i on machine M_k

Cp_i: Completion time of part P_i in processing

Cp_{ij}: Completion time of the j-th operation of part P_i

Cp_{ijk}: Completion time of the j-th operation of part P_i on machine M_k

En_i: The inventory time of part P_i

L: A given large positive integer

Sa_i: Starting time of part P_i in assembly

Ta_i: Operation time of part P_i in assembly

Ca_i: Completion time of part P_i in assembly

2.2. Building the objective functions for JSS-AP

To build the objective functions for JSS-AP, some assumptions are made as follows: any part can be processed at any time, there is no breakdown of the machine during the operation. The transportation time of each part is included in the operation time. Based on above assumption, the optimization objective functions for JSS-AP are given as follows:

Objective function 1: $Min\ F_1$

Where F_1 is the completion time.

$$F_1 = Ca_i,\ \text{when } P_i \text{ is the last part to be assembled} \quad (1)$$

Objective function 2: $Min\ F_2$

Where F_2 is inventory time of parts.

$$F_2 = \sum_{i=1}^{n} En_i \quad (2)$$

In the part processing, the completion time of the j-th operation of part P_i can be calculated as:

$$Cp_{ij} = Sp_{ij} + \sum_{k=1}^{m} X_{ijk} \times Tp_{ijk} \quad (3)$$

The completion time of part P_i in processing is given as:

$$Cp_i = Cp_{ij},\ \text{when } j{=}Q_i \quad (4)$$

In the assembly process, the completion time of part P_i in assembly can be calculated as:

$$Ca_i = Sa_i + Ta_i \quad (5)$$

The starting time of part P_{Ai} in assembly can be calculated as:

$$Sa_{Ai} = max\left(Cp_{Ai},\ Ca_{A(i-1)}\right) \quad (6)$$

Based on Eq.(4), Eq.(5) and Eq.(6), the completion time of part P_i in assembly can be concluded, and further, the completion time of the product can be concluded with Eq.(1).

Under a given an assembly sequence S: {$A1$, $A2$, ….., Ai, ….., An}, the inventory occupancy of part P_{Ai} will occur if all processing operations of part P_{Ai} are completed and the assembly of part $P_{A(i\text{-}1)}$ has not been completed. The inventory time of part Pi can be concluded with Eq.(7):

$$En_i = \max(Sa_i - Cp_i,\ 0) \quad (7)$$

Based on Eq.(7), the inventory time of parts during production can be obtained with Eq.(2).

2.3. Constraints

To solve the JSS-AP problem, the following constraints should be satisfied:

(1) Only one operation of a part can be processed at a time:

$$Cp_{ij_1} - Cp_{ij_2} + L \times (1 - Y_{ij_1j_2}) \geq Tp_{ij_1},\ \ \forall i, j_1, j_2 \quad (8)$$

Where,

$$Y_{ij_1j_2} = \begin{cases} 1 & \text{For part } P_i,\ \text{if the } j_2\text{-th operation is processed before the } j_1\text{-th operation} \\ 0 & \text{Else} \end{cases}$$

(2) Each machine can only process one operation at a time:

$$Cp_{i_1j_1k} - Cp_{i_2j_2k} + L \times (1 - Z_{i_1i_2k}) \geq Tp_{i_1j_1k},\ \text{if } X_{i_1j_1k} = X_{i_2j_2k} = 1, \forall i_1, i_2, j_1, j_2, k$$
(9)

Where,

$$X_{ijk} = \begin{cases} 1 & \text{If the } j\text{-th operation of part } P_i \text{ is processed on machine } M_k \\ 0 & \text{Else} \end{cases}$$

$$Z_{i_1i_2k} = \begin{cases} 1 & \text{If part } P_{i_2} \text{ is processed before part } P_{i_1} \text{ on machine } M_k \\ 0 & \text{Else} \end{cases}$$

(3) Each operation can only be processed on one machine at a time:

$$\sum_{k=1}^{m} X_{ijk} = 1,\ \forall i, j \quad (10)$$

(4) The assembly process of part P_i cannot start until all processing operations of part Pi are completed;

$$Cp_i \le Sa_i,\ \forall i \quad (11)$$

(5) The assembly process of part P_{Ai} cannot start until the assembly of part $P_{A(i-1)}$ is completed

$$Ca_{A(i-1)} \le Sa_{Ai},\ \forall i \quad (12)$$

2.4. Solving JSS-AP problem with NSGA-II

In order to solve the JSS-AP problem, an optimization approach with non-dominated sorted genetic algorithm-II (NSGA-II) is proposed in this section. According to the established mathematical optimization model of JSS-AP, an efficiency algorithm with the decision-making of the non-dominated solution is proposed to obtain the optimal solution of the JSS-AP problem.

2.4.1 Selection

In a group of solutions, if they are all non-dominated solutions, the Pareto level of the non-dominated solution is 1, then this group of non-dominated solutions is deleted from the total solution set, and the Pareto level 2 of the non-dominated solution is generated from the rest solutions according to the non-dominated relationship, and so on. The Pareto level of all solutions in the solution set can be obtained, as shown in Figure 1. According to the objective functions, the individuals of the same Pareto level are ordered. In a group of solutions, set F_1^{max}, F_2^{max} to be the maximum value of the objective function F_1, F_2, respectively, and F_1^{min}, F_2^{min} to be the minimum value of the objective function F1, F2, respectively. The n_d which represents the crowding distance of the two solutions on the boundary N_{d1} and N_{d2} is set as infinite. According to Eq.13, the crowding distance of the *i*-th solution can be obtained, where $F_m(i+1)$ and $F_m(i-1)$ represent the m-th objective function values of the $(i+1)$-th and $(i-1)$-th solution, respectively. N_{obj} is the number of the objective functions.

$$n_{d_i} = \sum_{m=1}^{N_{obj}} (F_m(i+1) - F_m(i-1)) / (F_m^{max} - F_m^{min}) \quad (13)$$

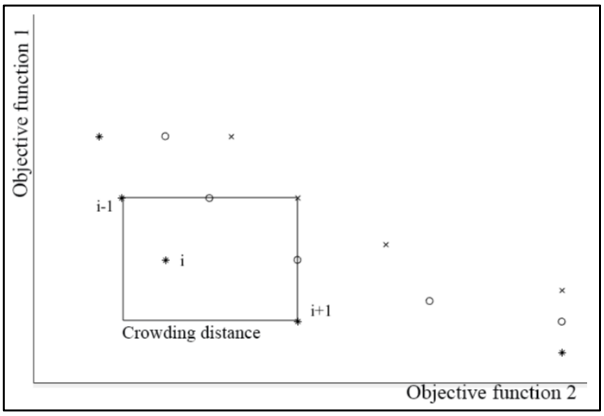

Figure 1 Pareto solution

IJAMSR 3 (5) www.ijamsr.com CrossRef: https://doi.org/10.31426/ijamsr.2020.3.5.3311

IJAMSR

International Journal of
Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281

Based on the non-dominated solution level and crowding distance, the traditional tournament method is used to select the solution. In each selection process, the non-dominated solutions with the lowest level and largest crowding distance are selected to prevent the local convergence of the algorithm.

2.4.2 Crossover

In crossover operation, a pair of chromosomes are randomly selected from the initial population as parents. Two cut points are randomly generated in the chromosomes representing the processing sequence, then two offspring are generated respectively by exchanging the gene fragments in two parent chromosomes.

2.4.3 Mutation

In mutation operation, two genes are randomly selected in each chromosome segment of the offspring, and then the two selected genes are exchanged in each offspring. The above mutation method can ensure the rationality of the offspring chromosome.

2.4.4 Decision making of non-dominated solution

After obtaining a set of non-dominated solutions for the JSS-AP problem, it is necessary to make a decision on this set of solutions and choose a solution more suitable for production needs. The membership degree and variance weighted sum are used to select the Pareto solutions as follows:

$$U^k = \frac{\sum_{i=1}^{Nobj} U_i^k \times \alpha_i}{\sum_{j=1}^{\mu} \sum_{i=1}^{Nobj} U_i^j \times \alpha_i} \quad (14)$$

$$U_{i=} \begin{cases} 1 & F_i \le F_i^{\max} \\ \dfrac{F_i^{\max} - F_i}{F_i^{\max} - F_i^{\min}} & F_i^{\min} \le F_i \le F_i^{\max} \\ 0 & F_i \ge F_i^{\max} \end{cases} \quad (15)$$

Where, U_k is the value of the k-th non-dominated solutions after decision-making calculation, μ is the number of non-dominated solutions, N_{obj} is the number of objectives, U_i^k is the value of the i-th objective of the k-th non-dominated solutions, U_i^j is the value of the i-th objective of the j-th non-dominated solutions, and is the weight of objective. According to the preference of different objectives, the weight value can be changed. F_i is the value of the i-th objective of non-dominated solutions, F_i^{max} and F_i^{min} are the maximum value and minimum value of the *i*-th objective of non-dominated solutions, respectively, U_i can be calculated according to Eq.(15). Then, the non-dominated solution set can be selected through comparing the value of U^k.

2.4.5 Framework of the proposed optimization approach to solve JSPCAP problem

Based on the technologies discussed above, the framework of the proposed optimization approach with NSGA- II to solve JSS-AP problem is shown in Figure 2.

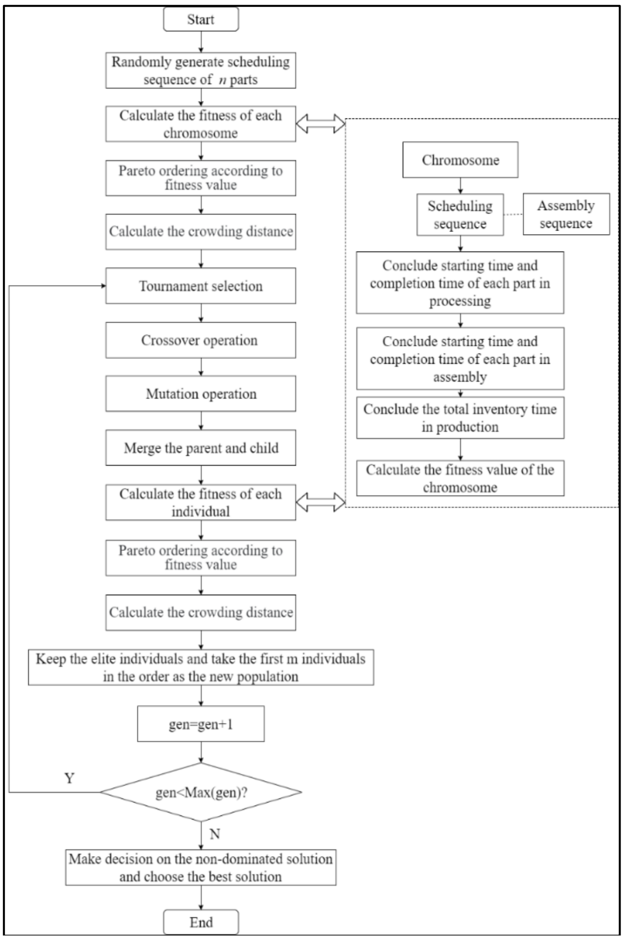

Figure 2 Framework of the optimization approach to solve JSS-AP problem.

3. Results and Discussion

3.1 The test of the proposed JSS-AP approach

To verify the proposed approach to solve JSS-AP problem, a product consisting of six parts is used as a case study. The processing information of each parts is given in Table 1, the assembly process information of each part is given in Table 2, and the given assembly

IJAMSR 3 (5) www.ijamsr.com CrossRef: https://doi.org/10.31426/ijamsr.2020.3.5.3311

International Journal of
Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281

sequence of the product is shown in Figure 3. In the proposed approach with NSGA-II, set the population size as 200, set the crossover probability as 0.8, set the mutation probability as 0.6, set the maximum generation as 200. In the decision-making process, there is no obvious preference between the two optimization objectives - completion time and inventory time, so the weights are set as 0.5 and 0.5, respectively.

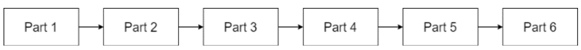

Figure 3 Assembly sequence of the product

Table 1 Processing information of each part

Part number	Operation	Processing time (h)	Machine
	$P_{1,1}$	40	M_2
1	$P_{1,2}$	43	M_3
	$P_{1,3}$	19	M_3
	$P_{2,1}$	53	M_2
2	$P_{2,2}$	22	M_1
	$P_{2,3}$	27	M_3
	$P_{3,1}$	53	M_2
3	$P_{3,2}$	22	M_1
	$P_{3,3}$	27	M_3
	$P_{4,1}$	21	M_4
4	$P_{4,2}$	53	M_2
	$P_{4,3}$	50	M_3
	$P_{5,1}$	53	M_3
5	$P_{5,2}$	29	M_2
	$P_{5,3}$	38	M_4
	$P_{6,1}$	19	M_1
6	$P_{6,2}$	27	M_3
	$P_{6,3}$	30	M_2

Table 2 Assembly process information of each part

Part number	1	2	3	4	5	6
Operation time (*h*)	16	16	16	32	16	16

Based on the above parameter setting, the best solution of NSGA-II running 200generation in one test is shown in Figure 4.

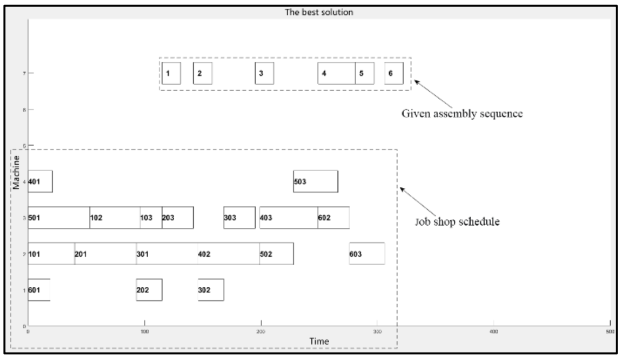

Figure 4 The best solution obtained by NSGA-II

As shown in Figure 4, the assembly process will start at about 110 hours after the start of part processing, and then the part processing process and assembly process will be carried out at the same time. By making decision on the non-dominated solution and choose the best solution, the completion time is 322h, and the inventory time of parts is 15h.

3.2 Comparison Test

For comparison, the JSS approach without considering inventory time is tested with the same case, with the NSGA-II using the same parameter setting, the best solution obtained in one test can be obtained. Table 3 shows the comparison between the results of the proposed JSS-AP approach and the JSS approach without considering inventory time. It can be seen from Table 3 that the completion time and inventory time obtained with the proposed JSS-AP approach, are both shorter than that obtained with JSS approach without considering inventory time, especially the inventory time can be greatly shortened.

Table 3 Comparison of the optimization results

Criterion	Proposed JSS-AP approach	JSS approach without considering inventory time
Completion time (*h*)	322	361
Inventory time (*h*)	15	279

4. Conclusion

This paper proposes an approach for job-shop scheduling considering the assembly process (JSS-AP) in production, where job shop scheduling is carried out according to the given assembly sequence, with the objective to minimize the completion time and the inventory time during production. The mathematical models including the fitness functions and the constraints are established for job shop scheduling considering the assembly process, and the non-dominated sorted genetic algorithm-II is applied to solve the problem, by which the decision-making of the non-dominated solutions is carried out according to the decision maker's preference for the objective in job-shop scheduling. The case studies show that compared with the JSS approach without considering inventory time, the obtained completion time and inventory time are both shorter than that obtained with JSS approach without considering inventory time, especially the inventory time can be greatly shortened. Therefore, the proposed JSS-AP approach can shorten the lead time and save the manufacturing cost of the enterprise more effectively.

References

1) *Gong G, Deng Q, Chiong R, Gong X, Huang H. An effective memetic algorithm for multi-objective job-shop scheduling. Knowledge-Based Systems, 2019. 182: 1-14.*
2) *Kurdi M. A new hybrid island model genetic algorithm for job shop scheduling problem. Computers & Industrial Engineering, 2015. 88: 273-283.*
3) *Wang YF. Adaptive job shop scheduling strategy based on weighted Q-learning algorithm. Journal of Intelligent Manufacturing, 2020. 31 (2): 417-432.*
4) *Yang F, Ye CM, Shi MH. A hybrid grey cuckoo search algorithm for job-shop scheduling problems under fuzzy conditions. Advances in Production Engineering & Management, 2018. 13 (3): 254-266.*

IJAMSR 3 (5) www.ijamsr.com CrossRef: https://doi.org/10.31426/ijamsr.2020.3.5.3311

International Journal of
Advanced Multidisciplinary Scientific Research (IJAMSR) ISSN:2581-4281

5) *Zhang J, Wang WL, Xu XL. A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility. Journal of Intelligent Manufacturing, 2017. 28 (8): 1961-1972.*
6) *Asadzadeh L. A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy. Computers & Industrial Engineering, 2016. 102:359-367.*
7) *Singh MR, Singh M, Mahapatra SS, Jagadev N. Particle swarm optimization algorithm embedded with maximum deviation theory for solving multi-objective flexible job shop scheduling problem. International Journal of Advanced Manufacturing Technology, 2016. 85 (9-12): 2353-2366.*
8) *Zhu Z, Zhou X. An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. Computers & Industrial Engineering, 2020. 140: 1-16.*
9) *Xiong W, Fu D. A new immune multi-agent system for the flexible job shop scheduling problem. Journal of Intelligent Manufacturing, 2018. 29(4): 857-873.*
10) *Caldeira RH, Gnanavelbabu A. Solving the flexible job shop scheduling problem using an improved Jaya algorithm. Computers & Industrial Engineering, 2019. 137: 1-16.*
11) *Ajchara P, Arit T. Memetic algorithm based on marriage in honey bees optimization for flexible job shop scheduling problem. Memetic Computing, 2017. 9(4): 295-309.*
12) *Dai M, Tang D, Giret A, Salido MA. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robotics and Computer Integrated Manufacturing, 2019. 59:143-157.*
13) *Fattahi P, Rad NB, Daneshamooz F, Ahmadi S. A new hybrid particle swarm optimization and parallel variable neighborhood search algorithm for flexible job shop scheduling with assembly process. Assembly Automation, online, 2020. DOI: 10. 1108/ AA-11-2018-0178.*
14) *Pereira MT, Santoro MC. An integrative heuristic method for detailed operations scheduling in assembly job shop systems. International Journal of Production Research, 2011. 49(20): 6089–6105.*
15) *Allahverdi A, Al-Anzi FS. The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. International Journal of Advanced Manufacturing Technology, 2008. 37(1-2): 166–177.*
16) *Lin WH, Deng QW, Han WW, Gong GL, Li KX. An effective algorithm for flexible assembly job-shop scheduling with tight job constraints. International Transactions in Operational Research, online, 2020. DOI: 10. 1111/ itor. 12767.*
17) *Zhang SC, Li X, Zhang BW, Wang SY. Multi-objective optimisation in flexible assembly job shop scheduling using a distributed ant colony system. European Journal of Operational Research, 2020. 283(2):441-460.*
18) *Liao CJ, Lee CH, Lee HC. An efficient heuristic for a two-stage assembly scheduling problem with batch setup times to minimize makespan. Computers & Industrial Engineering, 2015. 88: 317–325.*
19) *Du H, Liu DC, Zhang MH. A hybrid algorithm based on particle swarm optimization and artificial immune for an assembly job shop scheduling problem. Mathematical Problems in Engineering, 2016. DOI: 10. 1155/ 2016/ 3938679.*