Int j simul model 20 (2021) 2, 422-433
Original scientific paper

PRODUCTION MANAGEMENT OF MULTI-OBJECTIVE FLEXIBLE JOB-SHOP BASED ON IMPROVED PSO

Mao, C. L.
School of Intelligent Manufacturing, Nanyang Institute of Technology, Nanyang 473004, China
E-Mail: maocuili@nyist.edu.cn

Abstract
It is of great practical significance to improve the traditional particle swarm optimization (PSO) for the production management of multi-objective flexible job-shop. However, the current studies have not solved the problem that the traditional PSO cannot apply to the production and processing environment with numerous uncertain changes. Therefore, this paper improves the PSO for the production management of multi-objective flexible job-shop under different conditions. Firstly, the author modelled the production management of high-dimensional dynamic multi-objective flexible job-shop, and explained the pre-reaction dynamic rescheduling method for production management. Then, the PSO was improved in terms of inertial weight, learning factors, global search ability, and local search ability, and the dynamic response mechanism was presented for the improved algorithm. The feasibility of the improved PSO was demonstrated through experiments. The research provides a reference for applying the improved PSO to the optimization in other fields.

(Received in February 2021, accepted in May 2021. This paper was with the author 1 month for 2 revisions.)

Key Words: Multi-Objective Flexible Job-Shop Production Management, Improved Particle Swarm Optimization (PSO), Dynamic Response

1. INTRODUCTION

With the rising demand for more diverse products, the job-shop production management has evolved from the traditional single-machine single objective problem or multi-machine single objective model to the production control of various products in small batches [1, 2]. Since the production and processing environment contains many uncertain changes, the production manager must consider the production management of flexible job-shop from multiple angles, and solve this problem with closely correlated yet sometimes conflicting elements [3, 4]. The production management of the multi-objective flexible job-shop is to make a compromise between the multiple constraints on the objectives [5-7]. Represented by particle swarm optimization (PSO), evolutionary algorithms have been well applied to multi-objective optimization problems. It is of great practical significance to improve the traditional PSO for the production management of multi-objective flexible job-shop.

The traditional secluding methods perform poorly in production control over small batches of various customized products [8-11]. Danielsson et al. [12] combined the genetic algorithm (GA) with the PSO into a hybrid algorithm, and improved the hybrid algorithm with nearest neighbour search (NNS) algorithm. Ali et al. [13] optimized the particle velocity and position of the PSO based on the particle optimization model and the similarity between Cauchy distributions. Nouri et al. [14] predicted the start time and duration of machine maintenance based on Weibull distribution and machine states, constructed a production control model with minimum makespan, and solved the problem with a bi-PSO algorithm.

The production task sorting of an integrated job-shop with multiple production lines (MPLIJS) can be equivalent to a complex combinatory optimization problem. During the production planning, the job-shop should give fully consideration to the production technique and organization for the products [15-18]. Abdelmaguid [19] summarized the production task sorting and adjustment MPLIJS as the operation sequence update of the integrated job-shop with multi-channel buffer zones under the just-in-time production model, and took production

https://doi.org/10.2507/IJSIMM20-2-CO11

load balancing and material consumption levelling as the goals of job-shop production control. Yokoyama et al. [20] introduced the ideas of swarm library and simulated annealing (SA) algorithm to the improved discrete PSO, and let the particles in the library to lead the particles in other swarms in velocity and position updates. Ishigaki and Takaki [21] integrated migrating birds optimization (MBA) algorithm into the PSO: the local optimal of particles and global optimal of the swarm were updated by adjusting the degree that each independent particle senses the information of the other particles.

For job-shop production control, the complexity of the problem increases with the number of parameters and constraints [22-25]. He et al. [26] introduced the clonal selection strategy to the selection of scheduling rules for dynamic production control of flexible job-shop; elite selection was combined with roulette strategy to retain the best individuals, and obtain the set of feasible solutions. To ensure the uniform distribution of initial solutions and accelerate the algorithm speed, Li et al. [27] implemented heuristic initialization of the particle swarm, and improved the inertial weight of particle movement based on the shrinkage factor.

When the production and processing environment contains many uncertain changes, several questions need to be answered: whether the PSO has a good effect on production control; what should be done to prevent premature convergence. This paper improves the PSO for the production management of multi-objective flexible job-shop under different conditions. Section 2 models the production management of high-dimensional dynamic multi-objective flexible job-shop. Section 3 explains the pre-reaction dynamic rescheduling method for production management. Section 4 improves the PSO in terms of inertial weight, learning factors, global search ability, and local search ability. Section 5 describes the dynamic response mechanism for the improved algorithm. Experimental results confirm the feasibility of the proposed algorithm.

2. PROBLEM MODELLING

During the production control of flexible job-shop, the real-time control strategy needs to be updated at each triggering moment of rescheduling. Based on the current states on the machines and job operations in the production control system, it is necessary to select the most suitable machine for each operation of each job, so that the multiple pre-set objectives can be optimized. Let h_0 and h_k be the initial time and rescheduling time, respectively. At time h_k, a new management plan should be prepared in view of the available machines, the outstanding operations of the old task and the new task, as well as other relevant processing information. The objective function can be defined as:

$$minG = [g_1(h_k), g_2(h_k), g_3(h_k), g_4(h_k)]^T \tag{1}$$

where, $g_1(h_k)$, $g_2(h_k)$, $g_3(h_k)$, and $g_4(h_k)$ are four descriptive objectives of the new management plan. Let $M(h_k)$ be the jobs to be processed at time h_k; $JH_i(h_k)$ and $BE_i(h_k)$ be the delivery date and start time for the outstanding operations, respectively, at this time; $J_i(h_k)$ be the completion time of the i^{th} job. Then, the production control efficiency $g_1(h_k)$ can be calculated by:

$$g_1(h_k) = 5\left(\underset{i=1,2,\ldots,M(h_k)}{max} \left(T_i(h_k)\right) - \underset{i=1,2,\ldots,M(h_k)}{min} \left(BE_i(h_k)\right) \right) + 2\left(\sum \underset{i=1,2,\ldots,M(h_k)}{max} \left(0, T_i(h_k) - JH_i(h_k)\right) \right) \tag{2}$$

Eq. (2) shows that $g_1(h_k)$ contains two items: the difference between the earliest start time and latest end time of all the outstanding operations of the old task, and the total delay of all delayed jobs. This is because the outstanding operations include both the outstanding operations of the old task, and those of the new task. Let $SU_i(h_k)$ be the subscript of the first outstanding operation; $P_{ij}(h_k)$ be the j^{th} operation of the i^{th} job $J_i(h_k)$; $ZN_{ijl}(h_k)$ be the processing time of $P_{ij}(h_k)$ on machine $PE_l(h_k) \in PE_{ij}(h_k)$; $M^*(h_k)$ be the number of jobs being processed at time h_k; Ψ_i be the shrinkage factor of the delivery date, which depends on the attributes of $J_i(h_k)$. Then, the delivery date $JH_i(h_k)$ can be calculated by:

$$JH_i(h_k) = BE_i(h_k) + \Psi_i \sum_{j=1,2,\ldots,SU_i(h_k)+M_i^*(h_k)-1} ZN_{ij} \tag{3}$$

Let $N_{D\text{-}PEl}(h_k)$ be the number of outstanding operations assigned to $PE_l(h_k)$ at time h_k; $ZN^{SU\text{-}h}$ be the remaining processing time of the jobs being processed; $P_{u\text{-}PEl}(h_k)$ be the u^{th} operation assigned to $PE_l(h_k)$; $ZN_{u\text{-}PEl}(h_k)$ and $T_{u\text{-}PEl}(h_k)$ be the processing time and completion time of $P_{u\text{-}PEl}(h_k)$, respectively. Then, the machine load $g_2(h_k)$ can be expressed as:

$$g_2(h_k) = \max_{i=1,2,\ldots,M(h_k)} \left(\sum_{u=1}^{Mn_l(h_k)} ZN_{u-PE_l}(h_k) + \delta_l ZN^{SU\text{-}h} \right) \tag{4}$$

where, $\delta_l(h)$ is a binary function indicating if a machine is processing any operation; if yes, $\delta_l(h) = 1$; otherwise, $\delta_l(h) = 0$. It can be seen that $g_2(h_k)$ is the total time of the operations on the machines that can be selected to execute tasks. Let $s_{ij}(h_k)$ and $e_{ij}(h_k)$ be the start time and end time of $P_{ij}(h_k)$, respectively. Then, $g_3(h_k)$, which characterizes objective stability can be defined as:

$$g_3(h_k) = \sum_{P_{ij} \in RS} 1.5 | s_{ij}(h_k) - s_{ij}(h_{k-1}) | + \sum_{P_{ij} \in DS} | s_{ij}(h_k) - s_{ij}(h_{k-1}) | + \sum | e_{ij}(h_k) - e_{ij}(h_{k-1}) | \tag{5}$$

Eq. (5) shows that $g_3(h_k)$ gives priority to the outstanding operations of the old task. Delayed or early completion requires the enterprise to pay a certain default cost or storage cost. Therefore, $g_3(h_k)$ also considers the total time deviation of operations that are completed before or behind than schedule.

Next, the energy consumption objective was introduced to the construction process of objective functions. Let $N_l(h_k)$ be the number of available machines at time h_k; ε_l be the energy consumption of PE_l per unit time; $\delta_{ijl}(h_k)$ be the binary function indicating whether a machine is processing any job; $\gamma_{ijl}(h_k)$ be the binary function indicating whether a machine is idle. Then, the energy consumption $g_4(h_k)$ of a machine can be expressed as:

$$g_4(h_k) = \int_{h_k}^{\max(T_i(h_k))} \left(\sum_{i=1}^{M^*(h_k)} \sum_{j=1}^{M(h_k)} \sum_{l=1}^{N_l(h_k)} \varepsilon_l \delta_{ijl}(h_k) + \sum_{l=1}^{N_l(h_k)} BE \varepsilon_l \gamma_l(h_k) \right) dh \tag{6}$$

Eq. (6) shows that $g_4(h_k)$ consists of the energy consumed in processing model, and that consumed in idle model.

3. PRE-REACTION DYNAMIC RESCHEDULING

Based on every moment that triggers rescheduling, the dynamic job-shop production control process can be decomposed into static production control time windows reflecting the current state of the job-shop. Each static window needs to timely update the current state of the job-shop, which includes the set of available machines M_E, the set of unavailable machines M_{UE}, the set of completed operations PP_E, the set of outstanding operations of the old task PP_O, and the set of outstanding operations of the new task PP_N. Note that failed or repaired machines must be timely removed or added to M_E. If an operation of a job in PP_E, PP_O or PP_N is not executed at rescheduling triggering moments, then the subsequent operations of the job will not be arranged for execution. In addition, the selectable start time of each operation of a job in PP_E, PP_O or PP_N should also be considered. Suppose a total of $M_A(h_k)$ jobs have outstanding operations at the current trigger moment, and $N_A(h_k)$ machines are available for processing. Then, the selectable start time of each job can be calculated by:

$$BE_i(h_k) = max(h_k, T_i(SU_i(h_k)-1)), i = 1,2,\ldots, M_A(h_k) \tag{7}$$

Let $T_{L\text{-}PEl}(h_{k\text{-}1})$ be the completion time of the operations completed by machine $PE_l(h_k)$ at time h_k-1. Then, the selectable start time of each machine can be calculated by:

$$RT_i(h_k) = max(h_k, T_{L\text{-}PE_l}(h_{k-1})), l = 1,2,\ldots,N_A(h_k) \tag{8}$$

Let $P_{ij}(h_k)$ be the u^{th} operation on machine PE_l. Then, the start time $ZN_{ij}(h_k)$ of that operation can be calculated by:

$$ZN_{ij}(h_k)=\begin{cases}\max\left(h_k, T_{i(SU_i(h_k)-1)}, T_{L-PE_l}(h_{k-1})\right), j=SU_i(h_k), u=1\\ \max\left(h_k, T_{i(SU_i(h_k)-1)}, T_{(u-1)-PE_l}(h_k)\right), j=SU_i(h_k), u=2,\dots,N_{D-PE_l}(h_k)\\ \max\left(h_k, T_{i(j-1)}(h_k), T_{L-PE_l}(h_{k-1})\right), j=SU_i(h_k)+1,\dots,SU_i(h_k)+M(h_k)-1, u=1\\ \max\left(h_k, T_{i(j-1)}(h_k), T_{(u-1)-PE_l}(h_k)\right), j=SU_i(h_k)+1,\dots,SU_i(h_k)+M(h_k)-1, u=2,\dots,N_{D-PE_l}(h_k)\end{cases} \tag{9}$$

4. PRINCIPLE AND IMPROVEMENT OF PSO

The solving flow of our algorithm for that problem is illustrated in Fig. 1. Suppose there exists an M_D-dimensional space, and a particle swarm of the size m_G. For the i^{th} particle in the swarm, the velocity is described as an M_D-dimensional vector $S_i = \{s_{i1}, s_{i2}, \dots, s_{iMD}\}$, and the position as $A_i = \{a_{i1}, a_{i2}, \dots, a_{iMD}\}$. During the search for optimal solutions, two types of optimal positions can be obtained: the local optimal JB found by independent particles and the global optimal OS found by the swarm. The two optimal solutions can be respectively vectorized as $JB_i = \{JB_{i1}, JB_{i2}, \dots, JB_{iMD}\}$ and $OB_i = \{OB_{i1}, OB_{i2}, \dots, OB_{iMD}\}$, respectively. Let θ be the inertial weight; α_1 and α_2 be learning factors; ξ_1 and ξ_2 be two random numbers in the interval of [0, 1]. Then, the velocity of the i^{th} particle in the $(\Psi+1)^{\text{th}}$ iteration can be updated by:

$$s_{iM_D}^{(\psi+1)}=\theta\times s_{iM_D}^{(\psi+1)}+\alpha_1\times\xi_1\times\left(JB_{iM_D}-a_{iM_D}^{(\psi)}\right)+\alpha_2\times\xi_2\times\left(OB_{iM_D}-a_{iM_D}^{(\psi)}\right) \tag{10}$$

The position of the i^{th} particle in the $(\Psi+1)^{\text{th}}$ iteration can be updated by:

$$a_{iM_D}^{(\psi+1)}=a_{iM_D}^{(\psi)}+s_{iM_D}^{(\psi+1)} \tag{11}$$

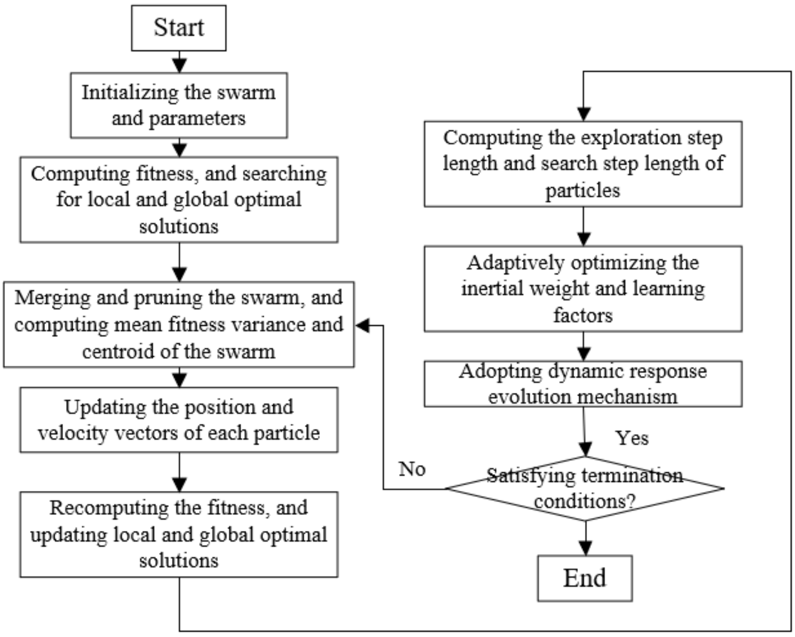

Start
Initializing the swarm and parameters
Computing fitness, and searching for local and global optimal solutions
Merging and pruning the swarm, and computing mean fitness variance and centroid of the swarm
Updating the position and velocity vectors of each particle
Recomputing the fitness, and updating local and global optimal solutions
Computing the exploration step length and search step length of particles
Adaptively optimizing the inertial weight and learning factors
Adopting dynamic response evolution mechanism
Yes
No
Satisfying termination conditions?
End

Figure 1: Solving flow of our algorithm for job-shop production control problem.

To iteratively reduce θ, this paper improves the parameters adaptively with the nonlinear decreasing function. Let ψ_{max} be the maximum number of iterations. Then, the improved θ can be expressed as:

$$\theta=\theta_{Max}-\left(\theta_{Max}-\theta_{Min}\right)\times\exp\left(\frac{1}{\psi_{\max}}-1\right)\times rand(0,1) \tag{12}$$

The learning factors α_1 and α_2 represent the information exchange between particles in the PSO. α_1 guides the particles moving towards the direction of local optimal particle, while α_2

guides the particles moving towards the direction of global optimal particle. Here, α_1 and α_2 change dynamically and adaptively by the loss function below:

$$\gamma(\mu,c)=\int_0^{\mu}\exp(-\psi)\psi^{c-1}d\psi \tag{13}$$

α_1 and α_2 can be respectively described by:

$$\begin{aligned}\sigma_1=\sigma_{max}&=\frac{(\sigma_{max}-\sigma_{min})}{\mu}\cdot\gamma\left(\mu,1-\frac{\psi}{\psi_{max}}\right)\\ \sigma_2=\sigma_{min}&=\frac{(\sigma_{max}-\sigma_{min})}{\mu}\cdot\gamma\left(\mu,1-\frac{\psi}{\psi_{max}}\right)\end{aligned} \tag{14}$$

To balance local search and global search of the PSO, the flower pollination algorithm (PFA) was introduced to modify the search model. After adaptive optimization, the switch probability *SP* can be described as:

$$SP=\frac{3}{4}+\frac{1}{4}\times rand(0,1) \tag{15}$$

Let *LF*(μ) be the selected Levy Flight function; ξ be a random number in the interval [0, 1]. Suppose $rand_{ND}$ is a random number obeying standard normal distribution. Then, the improved position of the ith particle in the (Ψ+1)th iteration can be updated by:

$$a_i^{(\psi+1)}=a_i^{(\psi+1)}+LF(\mu)\times\left(JB_{iM_D}-s_{iM_D}^{(\psi)}\right)+\xi\times\left(OB_{iM_D}-s_{iM_D}^{(\psi)}\right) \tag{16}$$

To prevent the PSO from falling into the local optimum trap in the late phase of iterations, the firefly algorithm was called to generate an elite solution *v*. If *v* is better than the current global optimal, the improved position of the ith particle in the (Ψ+1)th iteration can be updated by:

$$a_i^{(\psi+1)}=v+rand_{ND}\times\left(OB_{iM_d}-S_{iM_d}^{(\psi+1)}\right) \tag{17}$$

If the current local optimal is better than *v*:

$$a_i^{(\psi+1)}=a_i^{(\psi)}+rand_{ND}\times\left(JB_{iM_d}-S_{iM_d}^{(\psi+1)}\right) \tag{18}$$

Let *TD*(ψ) be the normal distribution function of small samples. The position vector can be initialized based on the elite solution *v*:

$$if\,(\psi>\psi_{\max})\{a_i^{(\psi+1)}=v+TD(\psi)\times\left(OB_{iM_d}-S_{iM_d}^{(\psi+1)}\right)\} \tag{19}$$

In case the particles move out of the boundary, the out-of-boundary particles were transferred into the search space of feasible solutions through boundary mutation optimization. Let $a^{(\psi)}{}_{pq}$ and a_B be the position of a particle and the global optimal position in the current iteration, respectively; *BD*(ψ) be the β distribution function; a_U and a_D be the upper and lower bounds of particle position, respectively. If the particle position surpasses a_U, the position can be updated by:

$$a_{pq}^{(\psi+1)}=a_U-BD(\psi)\times\left(a_{pq}^{(\psi)}-a_B\right) \tag{20}$$

If the particle position is below a_D, the position can be updated by:

$$a_{pq}^{(\psi+1)}=a_D+BD(\psi)\times\left(a_{pq}^{(\psi)}-a_B\right) \tag{21}$$

5. DYNAMIC RESPONSE MECHANISM

This paper proposes a dynamic response mechanism to the uncertain dynamic events based on centroid prediction and global optimal position. Firstly, the swarm diversity and distribution were updated by merging and pruning the particles. Fig. 2 shows the results of dynamic update of the particle swarm. The centroid of the updated swarm *ZQ* can be calculated by:

$$CP_{\psi}=\frac{1}{|ZQ_{\psi}|}\sum\nolimits_{a\in ZQ_{\psi}}a \tag{22}$$

The directional step length W_ψ from the particle position in the current iteration to the predicted position in the next iteration can be defined as:

$$W_\psi = CP_\psi - CP_{\psi-1} \tag{23}$$

Based on the known moving direction of the particle, the new position of the ith particle in the decision space can be predicted by:

$$a_i^{(\psi+1)} = a_i^{(\psi)} + W_\psi \tag{24}$$

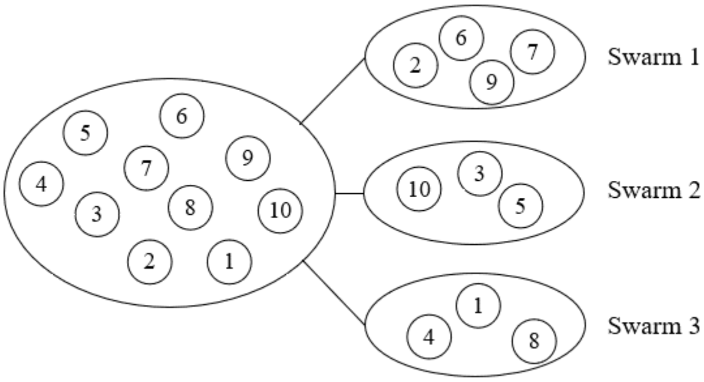


Figure 2: Results of dynamic update.

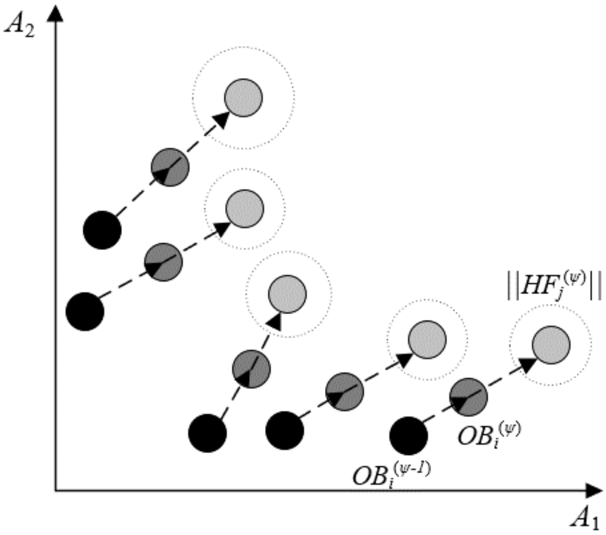


Figure 3: Exploration of the potential global optimal positions.

After predicting the new position, the particles belonging to different vector groups in the swarm were either retained or explored (Fig. 3). The step length for the exploration of potential global optimal positions can be obtained by:

$$DS_i^{(\psi)} = OB_i^{(\psi)} - OB_i^{(\psi\text{-}1)} \tag{25}$$

Then, the nearest particle $OB_j^{(\psi)}$ in the neighbourhood domain $NA\{h^i\}$ to particle $OB_i^{(\psi)}$ was searched for with the step length of:

$$HF_j^{(\psi)} = OB_j^{(\psi)} - OB_i^{(\psi)} \tag{26}$$

Once $DS_i^{(\psi)}$ and $HF_j^{(\psi)}$ had been acquired, it is assumed that the unit vector with random direction for improving particle disturbance is r'. Then, the potential particles generated through the exploration can be expressed as:

$$OB_i^{(\psi+1)} = OB_i^{(\psi)} + DS_i^{(\psi)} + rand(0,1)\left\|HF_j^{(\psi)}\right\|r' \tag{27}$$

Eq. (27) shows that the exploration of potential points faces the position vector of independent vectors rather than the whole swarm. This approach is more pertinent than the traditional random initialization of the swarm.

6. EXPERIMENTS AND RESULTS ANALYSIS

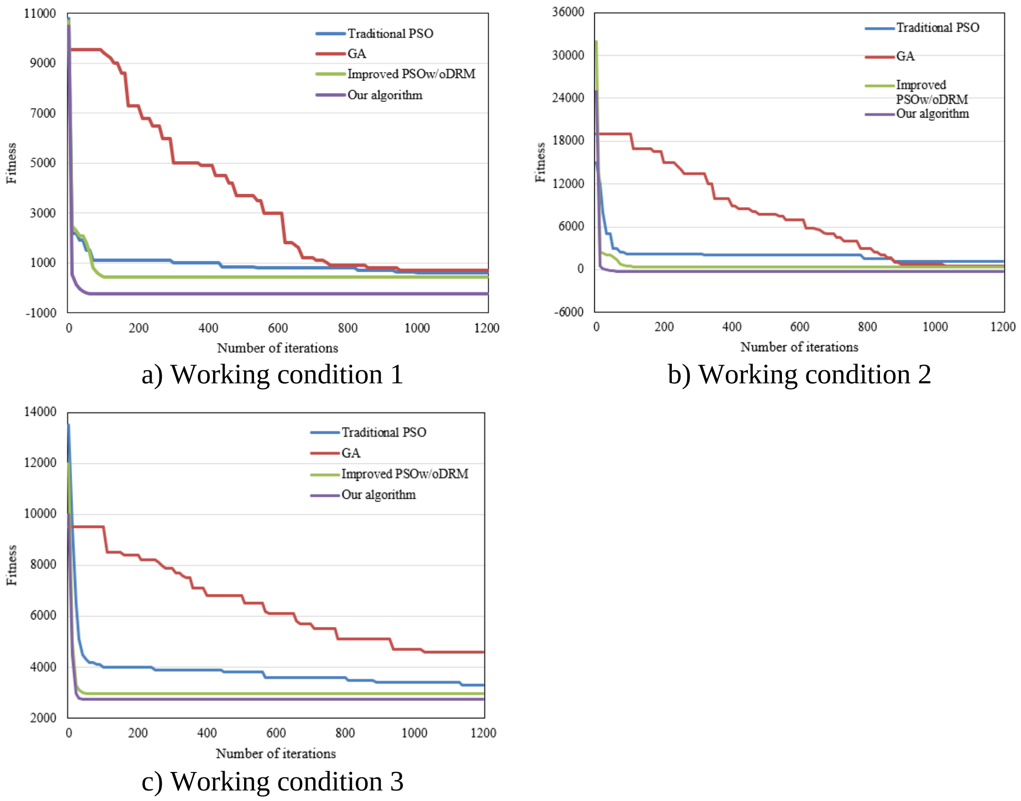
The dynamic response-based improved PSO was experimentally compared with the traditional PSO, GA, and improved PSO without dynamic response mechanism (improved PSOw/oDRM). During the experiments, the number of rescheduling triggers was set to 10, 20, and 30 in turn. Table I records the optimal solution, mean, number of iterations to convergence, and success rate of the plan of each of the four algorithms. As shown in Table I, our algorithm achieved better optimal solution and mean than other algorithms, and converged at the fastest rate. The success rate of our algorithm in obtaining the optimal production control plan approximated 100 %, much higher than that of any other algorithm. The above analysis shows that our algorithm has advantages in both convergence speed and accuracy.

Fig. 4 presents the convergence curves of the four algorithms under three different working conditions. It is intuitive that our algorithm was more efficient than the other algorithms in handling multi-objective production control problem. Fig. 5 shows the swarm diversity curve during the operation of our algorithm. It can be seen that the swarm diversity gradually declined before stabilizing at about 0.02. This promotes the balance between local and global searches during the convergence of the PSO.

Table II compares the production control results of different algorithms on high-dimensional dynamic multi-objective flexible job-shop. Our algorithm obtained the maximum under eight out of the ten working conditions. The overall performance of our algorithm is better than that of any other algorithm being tested. The means of the eight maximums were close to the optimal solutions, which further confirms the stability of our algorithm in optimization.

Table I: Data on the four algorithms under different conditions.

No. of working conditions	Dimensi-onality	Algorithm	Optimal solution	Mean	Number of iterations to convergence	Success rate
WC_1	2	Our algorithm	-1	-1	22	100 %
	2	Improved PSOw/oDRM	-1	-0.994	65	91 %
	2	GA	-1	-0.982	241	67.23 %
	2	PSO	-1	-0.956	318	43.18 %
WC_2	2	Our algorithm	-1	-1	25	100 %
		Improved PSOw/oDRM	-1	-0.996	63	92.3 %
	2	GA	-1	-0.976	17	75.39 %
	2	PSO	-1	-0.992	35	65.72 %
WC_3	2	Our algorithm	3	3	27	100 %
		Improved PSOw/oDRM	3	2.94	77	90.4 %
	2	GA	3	4.72	72	92.84 %
	2	PSO	3	5.91	91	79.65 %
WC_4	2	Our algorithm	3	3	26	100 %
		Improved PSOw/oDRM	3	2.85	63	92.3 %
	2	GA	3	5.26	55	83.71 %
	2	PSO	3	5.63	98	92.93 %



a) Working condition 1

b) Working condition 2

c) Working condition 3

Figure 4: Convergence curves under different working conditions.

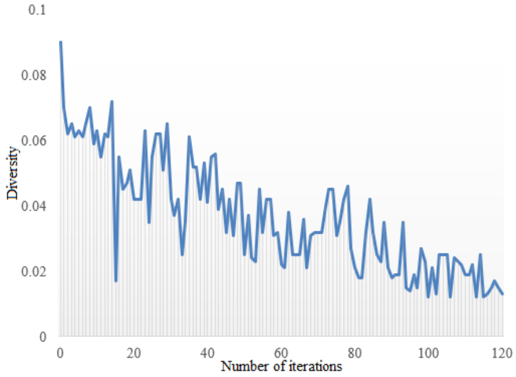
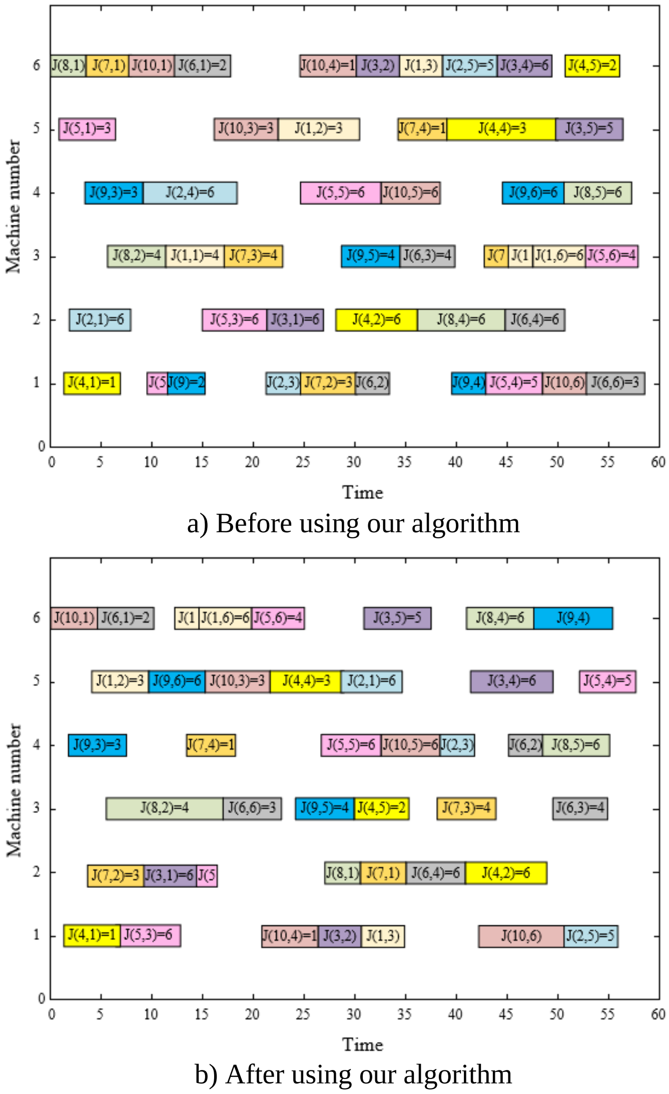


Figure 5: Change curve of swarm diversity.

Table II: Comparison of production control results on high-dimensional dynamic multi-objective flexible job-shop.

Serial no. of working condition	No. of jobs	No. of machines	Traditional PSO		GA		Improved PSOw/oDRM		Our algorithm	
			Optimal solution	Relative deviation	Optimal solution	Relative deviation	Optimal solution	Relative deviation	Optimal solution	Relative deviation
WC_1	9	5	41	0	42	0	42	2.7	45	41.2
WC_2	11	5	27	0	37	13.5	27	0	21	29
WC_3	14	7	203	0	208	1.9	209	0	208	315
WC_4	14	9	65	1.5	70	6.3	72	12.8	65	65.3
WC_5	14	3	172	2.2	192	8.2	178	3.6	173	175.2
WC_6	13	14	71	-3.5	87	27.5	67	2.3	68	68.9
WC_7	21	6	139	0.6	165	6.1	153	3.9	145	147.2
WC_8	21	11	536	0	539	0	521	0	539	535
WC_9	21	11	348	3.7	421	35.7	354	7.1	321	338.1
WC_{10}	21	16	239	-6.9	379	58.8	372	3.5	238	252.7

Figs. 6 a and 6 b present the Gantt charts of the flexible job-shop production control before and after using our algorithm. It can be seen that the optimization of the production control plan made the processing load more balanced on the machines, the operations more continuous, the completion time shorter, and the production control more reasonable.



a) Before using our algorithm

b) After using our algorithm

Figure 6: Gantt charts before and after the optimization of job-shop production control.

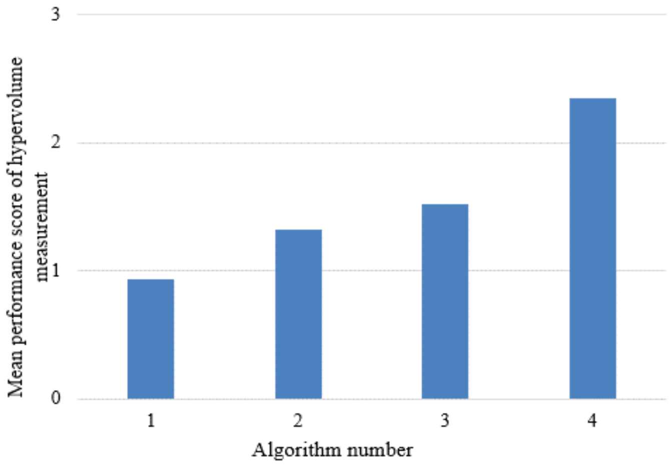


Figure 7: Hypervolume measurements by different algorithms at rescheduling triggering moments.

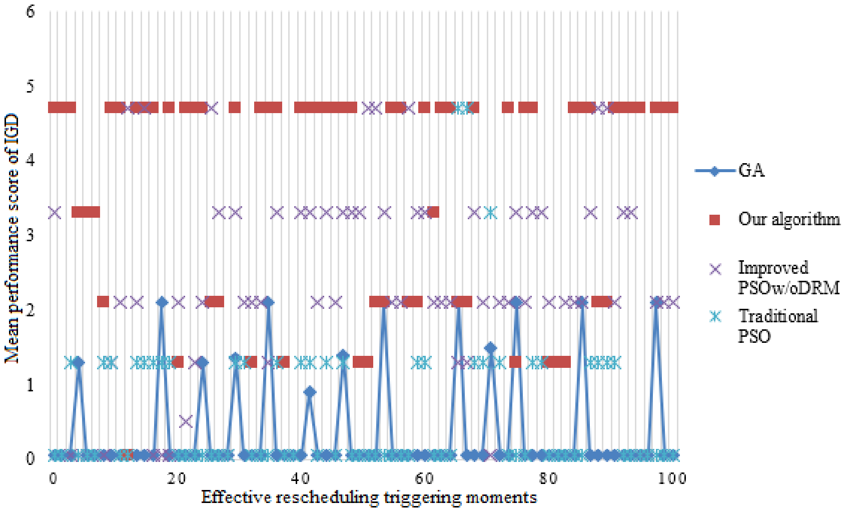


Figure 8: IGD measurements by different algorithms at rescheduling triggering moments.

Figs. 7 and 8 respectively display the hypervolumes and inverted generation distance (*IGD*) measured by different algorithms at rescheduling triggering moments. It can be inferred that, at most rescheduling moments, the improved PSO achieved much better hypervolume and *IGD* than the other algorithms. No algorithm could output the optimal control strategy at every triggering moment. However, our algorithm was the most likely to obtain the best result. It also appeared competitive in the composite ranking of hypervolume and *IGD*, and outperformed the other methods significantly at many moments on both measures.

7. CONCLUSIONS

This paper mainly improves the PSO for the production management of multi-objective flexible job-shop under different conditions. The superior convergence speed and accuracy of the proposed algorithm were confirmed through experiments on the production management of multi-objective flexible job-shop under different working conditions. Specifically, the solution quality and optimization stability of our algorithm were demonstrated through the comparison of production control results of other methods. The Gantt charts were adopted to prove that optimization of the production control plan made the processing load more balanced on the machines, the operations more continuous, and the completion time shorter. In addition, the hypervolumes and *IGD*s measured by different algorithms at rescheduling triggering moments were tested, which further reveal the advantages of our algorithm in the production management of high-dimensional dynamic multi-objective flexible job-shop.

ACKNOWLEDGEMENT

This paper was supported by National Natural Science Foundation of China (51875164), Henan Education Department Project (21B535002) and Henan Manufacturing Information Technology Service Center.

REFERENCES

[1] Zhu, J.; Shao, Z. H.; Chen, C. (2019). An improved whale optimization algorithm for job-shop scheduling based on quantum computing, *International Journal of Simulation Modelling*, Vol. 18, No. 3, 521-530, doi:10.2507/IJSIMM18(3)CO13

[2] Shivasankaran, N.; Senthil Kumar, P.; Raja, K. (2015). Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling, *International Journal of Computational Intelligence Systems*, Vol. 8, No. 3, 455-466, doi:10.1080/18756891.2015.1017383

[3] Caldeira, R. H.; Gnanavelbabu, A. (2019). Solving the flexible job shop scheduling problem using an improved Jaya algorithm, *Computers & Industrial Engineering*, Vol. 137, Paper 106064, 16 pages, doi:10.1016/j.cie.2019.106064

[4] Xiao, Y.; Li, C.; Song, L.; Yang, J.; Su, J. (2021). A multidimensional information fusion-based matching decision method for manufacturing service resource, *IEEE Access*, Vol. 9, 39839-39851, doi:10.1109/ACCESS.2021.3063277

[5] Zheng, F.; Wang, Z. (2019). Bi-objective flexible job shop scheduling with operation overlapping costs, *IFAC-PapersOnLine*, Vol. 52, No. 13, 893-898, doi:10.1016/j.ifacol.2019.11.308

[6] Mokhtari, H.; Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem, *Computers & Chemical Engineering*, Vol. 104, 339-352, doi:10.1016/j.compchemeng.2017.05.004

[7] Shan, D.; Wang, Y. (2019). A review of the application of genetic algorithms for solving flexible job shop scheduling problems, *Proceedings of the 2nd International Conference on Information Technologies and Electrical Engineering*, Paper 145, 5 pages, doi:10.1145/3386415.3387092

[8] Zahmani, M. H.; Atmani, B. (2019). A data mining based dispatching rules selection system for the job shop scheduling problem, *Journal of Advanced Manufacturing Systems*, Vol. 18, No. 1, 35-56, doi:10.1142/S0219686719500021

[9] Zahmani, M. H.; Atmani, B.; Bekrar, A.; Aissani, N. (2015). A real time data mining rules selection model for the job shop scheduling problem, *Proceedings of the 45th International Conference on Computers & Industrial Engineering 2015*, 465-472

[10] Ren, L.; Zhou, G.; Xia, F.; Fei, W.; Wu, C. (2015). A flexible job-shop scheduling under deteriorating processing time environment, *Metallurgical & Mining Industry*, Vol. 2015, No. 6, 184-192

[11] Beltran Ricaurte, I. I. B.; Londono Restrepo, M. I. (2015). Production scheduling in flexible job shop scheduling using an imperialist competitive algorithm (ICA) minimizing makespan, *Proceedings of the 10th Computing Colombian Conference*, 378-386, doi:10.1109/ColumbianCC.2015.7333449

[12] Danielsson, F.; Svensson, B.; Reddy, D. (2015). A genetic algorithm with shuffle for job shop scheduling problems, *Proceedings of the 29th Annual European Simulation and Modelling Conference*, 363-367

[13] Ali, A.; Hackney, P.; Bell, D.; Birkett, M. (2015). Genetic algorithms for solving bicriteria dynamic job shop scheduling problems with alternative routes, *Proceedings of the 2015 International Conference on Engineering*, Paper 31, 8 pages, doi:10.1145/2832987.2833038

[14] Nouri, H. E.; Driss, O. B.; Ghedira, K. (2015). A holonic multiagent model based on a combined genetic algorithm – tabu search for the flexible job shop scheduling problem, Bajo, J.; Hallenborg, K.; Pawlewski, P.; Botti, V.; Sanchez-Pi, N.; Duque Mendez, N. D.; Lopes, F.; Julian, V. (Eds.), *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Sustainability – The PAAMS Collection, PAAMS 2015, Communications in Computer and Information Science*, Springer, Cham, 43-54, doi:10.1007/978-3-319-19033-4_4

[15] Chaouch, I.; Driss, O. B.; Ghedira, K. (2019). A novel dynamic assignment rule for the distributed job shop scheduling problem using a hybrid ant-based algorithm, *Applied Intelligence*, Vol. 49, No. 5, 1903-1924, doi:10.1007/s10489-018-1343-7

[16] Xiao, S.-C.; Sun, S.-D.; Yang, H.-A. (2015). Hybrid estimation of distribution algorithm for stochastic job shop scheduling with earliness/tardiness penalty, *Control and Decision*, Vol. 30, No. 10, 1854-1860, doi:10.13195/j.kzyjc.2014.1269

[17] Bakhtar, S.; Jazayeriy, H.; Valinataj, M. (2015). A multi-start path-relinking algorithm for the flexible job-shop scheduling problem, *Proceedings of the 7th Conference on Information and Knowledge Technology*, 6 pages, doi:10.1109/IKT.2015.7288747

[18] Huang, M.; Fu, Y.; Wang, H.; Zhu, B.; Wang, X. (2015). Job-shop scheduling model and algorithm with machine deterioration, *Acta Automatica Sinica*, Vol. 41, No. 3, 551-558, doi:10.16383/j.aas.2015.c131067

[19] Abdelmaguid, T. F. (2015). A neighborhood search function for flexible job shop scheduling with separable sequence-dependent setup times, *Applied Mathematics and Computation*, Vol. 260, 188-203, doi:10.1016/j.amc.2015.03.059

[20] Yokoyama, S.; Iizuka, H.; Yamamoto, M. (2015). Priority rule-based construction procedure combined with genetic algorithm for flexible job-shop scheduling problem, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 19, No. 6, 892-899, doi:10.20965/jaciii.2015.p0892

[21] Ishigaki, A.; Takaki, S. (2017). Iterated local search algorithm for flexible job shop scheduling, *Proceedings of the 6th IIAI International Congress on Advanced Applied Informatics*, 947-952, doi:10.1109/IIAI-AAI.2017.126

[22] Li, X.; Lin, C.; Xu, X. (2019). A target tracking model for enterprise production monitoring system based on spatial information and appearance model, *Traitement du Signal*, Vol. 36, No. 4, 369-375, doi:10.18280/ts.360410

[23] Aathi, M. (2017). A novel algorithm for solving job-shop scheduling problem, *Mechanika*, Vol. 23, No. 4, 610-616, doi:10.5755/j01.mech.23.4.14055

[24] Bouazza, W.; Sallez, Y.; Beldjilali, B. (2017). A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect, *IFAC-PapersOnLine*, Vol. 50, No. 1, 15890-15895, doi:10.1016/j.ifacol.2017.08.2354

[25] Anaya Fuentes, G. E.; Hernández Gress, E. S.; Tuoh Mora, J. C.; Medina Marín, J. (2016). Solution of the job-shop scheduling problem through the traveling salesman problem, *Revista Iberoamericana de Automatica e Informatica Industrial*, Vol. 13, No. 4, 430-437

[26] He, Y.; Weng, W.; Fujimura, S. (2017). Improvements to genetic algorithm for flexible job shop scheduling with overlapping in operations, *Proceedings of the 16th International Conference on Computer and Information Science*, 791-796, doi:10.1109/ICIS.2017.7960100

[27] Li, H.; Cao, B.; Zhu, H. (2017). A variable neighborhood migrating birds optimization algorithm for flexible job shop scheduling, *International Journal of Performability Engineering*, Vol. 13, No. 7, 1020-1029