

Multi-objective Dynamic Scheduling Model of Flexible Job Shop Based on NSGAII Algorithm and Scroll Window Technology

Yingli Li and Jiahai Wang^(⊠)

School of Mechanical Engineering, Tongji University, Shanghai 201804, China jhwang@tongji.edu.cn

Abstract. The production process is often accompanied by a lot of disturbances, which make it difficult for flexible job shop to execute production according to the original job plan. It is necessary to dynamically adjust the production plan according to real-time conditions. To this end, this paper proposes a multi-objective dynamic scheduling model. In this model, scroll window technology and NSGAII algorithm is adopted to adapt the dynamic production evironment. A specific chromosome retention strategy and a variable objective selection mechanism are designed to ensure that the proposed model can select different objectives according to different disturbance events to solve the optimal solution. Finally, a case test is used to verify the feasibility and effectiveness of the model.

Keywords: Flexible job-shop · Dynamic scheduling · Evolutionary algorithm

1 Introduction

The flexible job-shop scheduling problem (FJSP), as an extension of the job-shop scheduling problem (JSP), has been proved to be a NP-hard problem [1]. FJSP is usually divided into two categories [2]: static scheduling and dynamic scheduling. Static scheduling is an optimized job plan generated by the management system based on the determined resource information, while dynamic scheduling is a response to the emergency in production process based on real-time production information. Compared with static scheduling, due to the uncertainty of dynamic scheduling, solving the problem of dynamic scheduling is more complicated. It is of great significance to study the flexible job shop dynamic scheduling problem (FJDSP), as described below.

On the one hand, the production process is often accompanied by the occurrence of emergencies. The timing of these events and their impact on production are often uncertain. This makes it difficult for the job shop to carry out production according to original plan. Therefore, compared with static scheduling, dynamic scheduling is more in line with the actual situation of job shop production control [3]. On the other hand, scheduling is the core functional modules of manufacturing execution system (MES), and its performance is directly related to the effect of MES. Therefore, the study of FJDSP is conducive to improve MES management level.

[©] Springer Nature Switzerland AG 2020

Y. Tan et al. (Eds.): ICSI 2020, LNCS 12145, pp. 435–444, 2020. https://doi.org/10.1007/978-3-030-53956-6_39

Due to its importance, many scholars paid much attention on FJDSP and established some dynamic scheduling models, including: genetic algorithm-based dynamic scheduling [4], particle swarm algorithm-based dynamic scheduling [5] and simulated annealing algorithm-based dynamic scheduling [6] and so on. However, the survey found that although many machining companies in China have deployed MES, they have not used the scheduling functions, and have only used resource management and monitoring modules. The reason is that the existing dynamic scheduling model has poor selfadaptability and cannot effectively respond to disturbances. The specific performance is that from static scheduling to dynamic scheduling, the optimization goal is single and fixed, which is the shortest maximum completion time, or the lowest penalty cost, or a combination of the two. This scheduling model with a single fixed objective will cause the algorithm ignoring other key issues when seeking the optimal solution, such as making a job unrestricted delay in order to achieve the shortest maximum completion time.

Aiming at the above existing problems of FJDSP, this paper uses NSGAII algorithm combined with scroll window technology to establish a multi-objective dynamic scheduling model DSNSGAII. This model can adaptively select optimization objectives for rescheduling according to different emergencies. This improves the reliability of production plan execution.

The rest of this paper is organized as follows: In Sect. 2, application scenarios is described. DSNSGAII is introduced in Sect. 3. In Sect. 4, case verification is presented. Section 5 are the conclusions.

2 Problem Description

The research object of this paper is flexible job shop, which characteristics are described as follows: There are N_i ($i \in (1, 2, ..., n)$) jobs to be produced on M machines, where iis the index of job number. Each job has j operations, O_{ij} represents the *jth* operation of job i. There is a predefined sequence between any two-adjoining operations of each job. Each operation can be processed by multiple machines. $m_{ij}(m_{ij} \subset M)$ represents the candidate machine set of O_{ij} . Each machine can only process one operation at a time, and one operation can only be processed by one machine at a time. There are two subproblems, one is how to assign operations to suitable machine, and the other is to find optimal production sequence for all operations assigned to machines. The optimization objectives used in this paper mainly include the following.

(1) Minimize makespan:

$$f_1(t) = \min\{\max(C_{ij})\}.$$
(1)

(2) Minimize total tardiness:

$$f_2(t) = \min \sum_{i=1}^{n} \left[\max\left(0, C_i - T_i^J\right) L\left(N_i^{y1}\right) + \max\left(0, C_i - T_i^{yJ}\right) L\left(N_i^{y2}\right) \right].$$
(2)

$$\begin{cases} T_i^x \le T_i^J, L\left(N_i^{y1}\right) = 1, \ L\left(N_i^{y2}\right) = 0 \ if \ L(N_i) = 1\\ T_i^x \le T_i^{yJ}, L\left(N_i^{y1}\right) = 0, \ L\left(N_i^{y2}\right) = 1 \ if \ L(N_i) = 0 \end{cases}$$
(3)

(3) The shortest delay time for a single job:

$$f_3(t) = \min\{\max(0, C_k - D_k)\}.$$
 (4)

 C_{ij} represents the completion time of operation O_{ij} . $L(N_i)$ is the delay tag. T_i^x is the completion time of job *i* in the new plan. T_i^J is the actual completion time of job *i* before dynamic scheduling. T_i^{yJ} is the original planned completion time of job *i*. C_k is the completion time of job *k*. D_k is the due date of job *k*.

3 DSNSGAII Model

3.1 Job Processing

According to the scroll window technology [7], when dynamic scheduling is executed, the state of the operation set may have four, namely: finished processing, being processing, waiting processing and waiting scheduling, as shown in Fig. 1. For the operation sets that have been processed, it can be removed when performing rescheduling. For the operation sets that is waiting for processing and waiting for scheduling can be retained. The key problem lies in how to deal with the operation sets that are being processed. In this paper, whether the dynamic scheduling is triggered by machine failure, the operation set being processed is divided into two cases to deal with separately.



Fig. 1. Job status diagram based on scroll window technology.

(1) Dynamic scheduling is not trigged by machine failure

In this case, the operation set of being processed is still completed by current machine. Its effect on dynamic scheduling is that the available time of this machine does not start from the dynamic scheduling trigger time, but the end time of the operation set of being processed. Its mathematical expression is shown in formula (5), where T_{MP} is the available start time of machine *P* after dynamic scheduling is triggered, t_c is the dynamic scheduling trigger time, t_{ijMP} is the required time

to complete the operation set of being processed when the dynamic scheduling is triggered.

$$T_{MP} = t_c + t_{ijMP}.$$
 (5)

(2) Dynamic scheduling is trigged by machine failure

In this case, the operation set of being processed cannot continue processing on the original machine, so it needs to participate in dynamic scheduling. Its impact on dynamic scheduling is that part of the products in the operation set have been completed. Therefore, the processing time required by the operation set should be the original planned time minus the time consumed by the completed products. Its mathematical expression is shown in formula (6), where T_{ijx} is the time required to complete the new operation set after removing the finished product, T_{ij} is the required processing time of O_{ij} of a product. Q_i is the product quantity of the operation set, and q_i is the finished product quantity in the operation set.

$$T_{ijx} = T_{ij}(Q_i - q_i).$$
(6)

3.2 Encoding and Decoding

Encoding: The A-B string coding method that used by Zhang et al. is adopted [8]. The chromosome includes two parts: machine selection and operation sequence. The total length of chromosome is the twice sum of all operations. as shown in Fig. 2.



Fig. 2. Chromosome coding

In the machine selection section, machine numbers are arranged according to operation numbers. The number represents the location of the machine in the candidate machine set. For example, the second number 1 in the machine selection part represents the machine position number corresponding to the operation O_{12} . That is, the machine located in the first position is selected from the candidate machine set of operation O_{12} , and the machine selected at this time is M2. In the operation sequence, the number represents the job number, and the number of occurrences represents the corresponding operation number. For example, the first appearance of 1 represents operation O_{11} , and the second appearance of 1 represents operation O_{12} . **Decoding:** It has been proved that the optimal solution is often within the set of active solutions [9]. In order to obtain the active solutions, the left shift interpolation method is adopted. The detail calculation method of left shift interpolation can reference to literature [8].

3.3 Genetic Manipulation

Population Initialization. In order to ensure the diversity of population, initial population is generated by the mixed method. Operation sequence is generated by mixing four methods: random generation method, shortest processing time priority rule, maximum remaining operation rule, and maximum remaining load rule. The respective proportions are: 50%, 20%, 20%, and 10%. The machine selection is generated by mixing three methods: random generation method, shortest processing time priority rule, and machine workload minimum priority rule. The respective proportions are: 50%, 30%, and 20%.

Chromosome Crossing. In order to make good individual genes inherited to offspring, a superior chromosome repetition crossover strategy is proposed. Step1: Build superior chromosome memory with capacity R. Step2: The chromosome with superior objective value are added to the chromosome memory. If the memory reaches the maximum capacity, the better chromosome replaces the poor one. Step3: An individual is selected in turn from the chromosome memory, and it is crossed with multiple non-memory chromosomes to obtain offspring. In addition, the operation of randomly selecting two chromosomes from all chromosomes for crossover remains. The two kinds of crossover methods account for 40% and 60%. The specific cross-operation adopts the IPOX and MPX methods proposed by Wang et al. [10].

Chromosome Mutation. Machine chromosome mutation uses a single point replacement method, which randomly selects one point from the chromosome as the mutation point and replaces the machine at the mutation point with the shortest processing time's machine in the candidate machine set. Operation chromosome mutation adopts two methods: double-point exchange and clockwork rotation, as shown in Fig. 3. Two-point exchange is to randomly select two different points from the operation chromosome as the exchange points and exchange the genes of these two points. The clockwork rotation mutation method is described as follows: randomly generate an integer n ($1 \le n \le L$), all genes in the operation sequence are moved n positions in a clockwise direction, and maintain the relative position invariant. The proportion of the two methods in chromosome mutations is 70% and 30% of the mutation size, respectively.

3.4 Chromosome Selection

There are generally three disturbances in job shop: normal order addition, urgent order insertion, and machine failure. Different disturbances have different requirements for rescheduling results. Therefore, these three cases are processed separately.



Fig. 3. The clockwork rotation mutation method

Normal Order Addition. In order to avoid the impact of new job addition on original job delivery time, a restriction rule is set. i.e. the original job was completed on time and still completed on time. The delay time of original delayed job in new plan cannot exceed the original delayed time. Thus, there are two objectives, one is that the original jobs have the shortest delay time in new plan, namely f_2 , and the other is minimum the makespan, namely f_1 . If there are individuals with f_2 of 0, add the individuals to next iteration and the chromosome memory. For other individuals, the chromosome is selected by fast non-dominant sorting and crowding distance calculation [11]. Finally, the solution with $f_1 = \min$ and $f_2 = 0$ is the optimal solution.

Insert Urgent Job. It is necessary to ensure that the urgent job is completed on time. In this case, the first objective is the shortest delay time for urgent job, namely f_3 . The second objective is to minimize the total delay time of all jobs. In addition, to ensure that all jobs are completed quickly, this section sets the auxiliary objective, which is minimize makespan. If there are individuals with f_3 of 0, add the individuals to next generation population and the chromosome memory. For other individuals, the chromosome is selected by fast non-dominant sorting and crowding distance calculation. When outputting the optimal solution, the hierarchical selection method is used. First, f_3 must be the smallest and the screening result is the population is $q_2(q_1 \subset q_2)$; finally, the individuals $q_3(q_3 \subset q_2)$ with the smallest f_1 is selected.

Machine Failure. The dynamic scheduling triggered by machine failure has a great impact on the original job plan, mainly because the machine is production resource, the reduction of production resource may lead to the delay of multiple jobs. In order to mitigate the impact of machine failure on production plan, the shortest total delay time f_2 was taken as objective. In order to deal with the case that machine failure did not cause job delay, take minimum makespan as the second objective. The chromosome with the smallest objective function value (min $f_2(f_2 \neq 0)$ or min $f_1(f_2 = 0)$) is added to chromosome memory. The population that enters the next iteration is selected by fast non-dominated sorting and calculation of the crowded distance. Finally, the solution with min $f_2(f_2 \neq 0)$ or min $f_1(f_2 = 0)$ is the optimal solution.

4 Case Verification

To test the effectiveness of DSNSGAII, the benchmark case ka 4X5 [12] is adopted, the information of this case is shown in Table 1. J_5 is the job of normal additional

Job no.	0 _{ij}	Processing time per machine						D _k
		M1	M2	M3	M4	M5		
J ₁	O ₁₁	2	5	4	1	2	20	190
	O ₁₂	5	4	5	7	5		
	O ₁₃	4	5	5	4	5		
J ₂	O ₂₁	2	5	4	7	8	20	240
	O ₂₂	5	6	9	8	5		
	O ₂₃	4	5	4	54	5		
J ₃	O ₃₁	9	8	6	7	9	10	120
	O ₃₂	6	1	2	5	4		
	O ₃₃	2	5	4	2	4		
	O ₃₄	4	5	2	1	5		
J_4	O ₄₁	1	5	2	4	12	30	100
	O ₄₂	5	1	2	1	2		

Table 1. The information of benchmark case.

and emergency insertion, its information shown in Table 2. The time of normal order addition, urgent order insertion and machine failure is generated randomly. The failure machine is also random selected.

Due to the optimization objectives in DSNSGAII is changed with the disturbances, it is hard to compare with other excellent algorithms directly, this paper only compares dynamic scheduling results with the classical genetic algorithm (MGA), which aims to minimize makespan and minimize total delay time. In the MGA, the weight of the two objectives is 0.6 and 0.4. In order to ensure the comparability of results, the DSNSGAII and the MGA adopt the same genetic methods, and the parameters adopts the same value: the population size is 200, crossover rate is 0.8, mutation rate is 0.8 and iterations is 100.

Job no.	0 _{ij}	Processing time per machine						D _k
		M1	M2	M3	M4	M5		
J ₅	O ₅₁	2	4	3	5	2	20	230
	O ₅₂	3	5	7	2	4		
	O ₅₃	4	6	4	3	5		

Table 2. The information of urgent insertion order.

On-time delivery is a guarantee to ensure customer satisfaction and improve enterprise credibility. Therefore, this paper takes the delay time as the evaluation index of the algorithm's advantages and disadvantages. The fewer the delay orders and the shorter the total delay time, the better the algorithm.



Fig. 4. The gantt chart of job-shop production plan

The experiment results are shown in Fig. 4. Figure 4(a) is the original production plan, as can be seen from this figure, dynamic scheduling is triggered at time 40, and only J₄ is delayed and its delay time is 30.

Normal order addition. After J_5 is added normally, the production plan obtained by DSNSGAII is shown in Fig. 4(b). Compared with Fig. 4(a), it can be seen that the original job was completed on time and still completed on time, such as J_1 , J_2 and J_3 . The original job is delayed, the new completion time of the job is no later than the original completion time, such as J_4 . As can be seen from Fig. 4(b), only two job is delayed, and the total delay time is 40. The production plan obtained by MGA is shown in Fig. 4(d). It can be seen that due to the objective weights are difference, the optimal solution is more inclined to the objective with a large weight. Although the maximum completion time of the results obtained by MGA is 220 and it is smaller than DSNSGAII, the three jobs of J_1 , J_3 and J_4 are delayed, and the total delay time is 190. This result not only leads to an increase of delay cost, but also affects customer satisfaction. Therefore, through comparison, we conclude that DSNSGAII is superior to MGA in normal order addition.

Urgent order insertion. After J_5 is added to production in the form of an urgent job, the production plan obtained by DSNSGAII is shown in Fig. 4(c). As can be seen from this figure, J_5 can complete on time, the three jobs of J_1 , J_3 and J_4 are delayed, the total delayed time is 120. Compared with Fig. 4(d), the DSNSGAII has fewer delay orders and the total delay time while ensuring that urgent job is completed on time. So, it is concluded that DSNSGAII is superior to MGA algorithm in terms of emergency order insertion.

Machine failure. Figure 4(e) and Fig. 4(f) are Gantt chart of the production plan solved by MGA and DSNSGAII after machine M4 fault. From Fig. 4(e), it can be seen that MGA will give priority to minimize makespan when solving. At the same time, the optimization goal with the smallest total delay time is weakened, resulting in a delay of 60 for J_3 and 30 for J_4 . The DSNSGAII can effectively balance the two goals. The maximum completion time is 220, which is the same as the result obtained by MGA. However, in the term of minimize total delay time, DSNSGAII only one job is delayed, and the total delay time is 30. Therefore, it is concluded that the DSNSGAII is more effective than MGA.

5 Conclusions

Dynamic scheduling is important for flexible job-shop production management. For solving dynamic scheduling effectively, this paper proposed a DSNSGAII model based on scroll windows technology and NSGAII algorithm. In the proposed model, a new crossover operator is designed to retain more good genes. A clockwork rotation mutation operator is adopted to generate good individuals. For different disturbances, different environment selection mechanism is designed to obtain the required optimal solution. Final, a case is given to verify the validity of the model.

Acknowledgement. This research is supported by the school-enterprise cooperation project of Tongji University under Grant No. kh0100020192160.

References

- Mohan, J., Lanka, K., Rao, A.N.: A review of dynamic job shop scheduling techniques. Procedia Manuf. (2019). https://doi.org/10.1016/j.promfg.2019.02.006
- Zambrano Rey, G., Bekrar, A., Prabhu, V., Trentesaux, D.: Coupling a genetic algorithm with the distributed arrival-time control for the JIT dynamic scheduling of flexible job-shops. Int. J. Prod. Res. (2014). https://doi.org/10.1080/00207543.2014.881575

- Renna, P.: Job shop scheduling by pheromone approach in a dynamic environment. Int. J. Comput. Integr. Manuf. (2010). https://doi.org/10.1080/09511921003642170
- Kundakci, N., Kulak, O.: Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. Comput. Ind. Eng. (2016). https://doi.org/10.1016/j.cie.2016. 03.011
- Al-Behadili, M., Ouelhadj, D., Jones, D.: Multi-objective particle swarm optimisation for robust dynamic scheduling in a permutation flow shop. In: Madureira, A.M., Abraham, A., Gamboa, D., Novais, P. (eds.) ISDA 2016. AISC, vol. 557, pp. 498–507. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53480-0_49
- Naderi, B., Azab, A.: An improved model and novel simulated annealing for distributed job shop problems. Int. J. Adv. Manuf. Technol. (2015). https://doi.org/10.1007/s00170-015-7080-8
- Nelson, R.T., Holloway, C.A., Mei-Lun Wong, R.: Centralized scheduling and priority implementation heuristics for a dynamic job shop model. AIIE Trans. (1977). https://doi.org/10. 1080/05695557708975127
- Zhang, G., Sun, J., Liu, X., Wang, G., Yang, Y.: Solving flexible job shop scheduling problems with transportation time based on improved genetic algorithm. Math. Biosci. Eng. 16, 1334– 1347 (2019). https://doi.org/10.3934/mbe.2019065
- Gao, J., Sun, L., Gen, M.: A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Comput. Oper. Res. (2008). https://doi.org/10.1016/j. cor.2007.01.001
- Wang, X., Gao, L., Zhang, C., Shao, X.: A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. Int. J. Adv. Manuf. Technol. (2010). https://doi.org/10.1007/s00170-010-2642-2
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. (2002). https://doi.org/10.1109/4235. 996017
- Kacem, I., Hammadi, S., Borne, P.: Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Math. Comput. Simul. (2002). https://doi.org/10.1016/S0378-4754(02)00019-8