NSGA-III for solving dynamic flexible job shop scheduling problem considering deterioration effect

Xiuli Wu*, Jing Li†, Xianli Shen†, Ning Zhao†

1Department of Logistics Engineering, School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, People’s Republic of China

Abstract: The production process of manufacturing systems is usually not static and always interrupted by stochastic events, such as the deterioration effect of cutting tools. This study focuses on the dynamic flexible job shop-scheduling problem considering the deterioration effect (DFJSP-DE). Two types of disturbances are considered, i.e. the implicit disturbance caused by the deterioration effect and the explicit disturbance caused by the reprocessing of unqualified jobs. Firstly, a step-deterioration effect model is proposed, with which the actual processing time of each operation can be predicted more accurately. A multi-objective optimisation model is formulated for the DFJSP-DE. The makespan, energy consumption, and stability of rescheduling solutions are three objectives to be optimised simultaneously. For this non-deterministic polynomial (NP)-hard problem, the non-dominated sorting genetic algorithm III is employed to search the Pareto solutions for the DFJSP-DE. Finally, the results of three numerical experiments show that the proposed approach can solve DFJSP-DE effectively and efficiently.

Nomenclature

| Symbol | Description |
|--------|-------------|
| n      | job quantity |
| j, g   | job index, j, g = 1, 2, ..., n |
| i, h   | operation index, i, h = 1, 2, ..., n_j |
| n_j    | operation quantity of job j |
| m      | machine quantity |
| k      | machine index, k = 1, 2, ..., m |
| t_r    | rescheduling time point |
| n(t_l) | quantity of jobs that have not completed all operations at t_l |
| n_j(t_l) | quantity of the unprocessed operations of job j at t_l |
| a(t_l) | index of the reprocessing job at t_l |
| b_j(t_l) | index of the reprocessing operation of the reprocessing job a at t_l |
| O_i,j  | ith operation of job j |
| O_i,j,t | event that operation O_i,j is processed on machine k |
| p_i,j,k | basic processing time of O_i,j,k |
| p_i,j,k,t | actual processing time of O_i,j,k |
| p_i,j,k, t | time for tool positioning and cutter’s approach to job j to process O_i,j,k on machine k |
| p_i,j,k, t | cutting time of O_i,j,k |
| b_i,j,k | time for cutter’s departure from job i after processing O_i,j,k |
| b_k | deterioration rate of machine k |
| t^min_k | lower bound of the deterioration duration threshold on machine k |
| t^max_k | upper bound of the deterioration duration threshold on machine k |
| C_max | makespan, i.e. the maximum completion time of a scheduling solution |
| W | stability of a rescheduling solution |
| S_i,j,k | starting time of O_i,j,k |
| C_i,j,k | completion time of O_i,j,k |
| C_i,j | completion time of O_i,j |
| X_i,j,k | X_i,j,k = 1 if O_i,j is processed on machine k; otherwise, X_i,j,k = 0 |

1 Introduction

Scheduling plays an important role in manufacturing systems. It deals with the resource allocation for tasks within a given time period, aiming at optimising one or more objectives [1]. Since the 1950s, many studies have been carried out on scheduling problems. As a result, some important models have been developed, e.g. single machine scheduling problems, parallel machine scheduling problems, flow shop scheduling problems, job shop scheduling problems (JSP) etc. Recently, in order to quickly respond to market demand, flexible manufacturing systems (FMS) have been
employed in many factories. In the FMS, machines are usually multi-function machines and can process different operations. As the extension of the JSP, the scheduling problem in FMS has been modelled as the flexible job shop-scheduling problem (FJSP).

In practical production, the processing environment is not static and often interrupted by stochastic events. These disturbances can be divided into explicit disturbances and implicit disturbances. A single explicit disturbance such as machine failure, the random arrival of a new job, and the reprocessing of an unqualified job, will have a significant impact on the production process. Different from the explicit disturbance, a single implicit disturbance may not affect the production process significantly. However, when the implicit disturbance increases with time and frequency, its accumulation will also affect the processing significantly, e.g. the processing time of jobs will be lengthened due to the deterioration effect of cutting tools. Furthermore, in some manufacturing systems, machining precision directly influences the quality of products, but cannot always be qualified. Also, the unqualified operations are required to be reprocessed immediately, thus avoid delivering the defective jobs to the subsequent processing stage and causing more repairing costs. Dynamic scheduling is necessary to respond to these disturbances, which is of great importance for the successful execution of the practical production process [2].

Up to now, many dynamic scheduling strategies have been developed, i.e. completely reactive scheduling, predictive-reactive scheduling, and pro-active scheduling [3, 4].

Completely reactive scheduling is also named as online scheduling. When a disturbance occurs in the process of online scheduling, a scheduling strategy is needed to be made according to the information obtained at the decision time [2]. For example, Lee et al. [5] considered both online scheduling and offline scheduling in a flow shop with the objective of minimising the makespan and they designed a greedy algorithm. Hopf et al. [6] studied an online flow shop-scheduling problem and developed a competitive deterministic online algorithm and a matching lower bound for online scheduling. Wang et al. [7] considered the online scheduling problem of minimising the makespan on unbounded parallel-batch machines. Lan et al. [8] studied the online scheduling of parallel machines with buffers and online deadline scheduling. Wu et al. [9] studied a framework of data driven scheduling method for the semiconductor production line and applied a machine-learning algorithm to obtain the dynamic scheduling model.

Predictive-reactive scheduling first generates an initial scheduling solution without considering any dynamic events. With time, once a disturbance occurs, the unexecuted operations up to that time are rescheduled. Predictive-reactive scheduling is the most widely used [10]. For example, Adib et al. [11] proposed an adaptive variable threshold solution to the job shop scheduling problem. Gao et al. [12] designed a two-stage artificial bee colony algorithm to solve the dynamic FJSP (DFJSP) problem considering new job insertion. Kundakci and Kulak [13] proposed a hybrid genetic algorithm for minimising makespan in a dynamic job shop-scheduling problem. Liu and Zhou [14] studied the dynamic scheduling for the unexpected arrival of a rush job in an open shop. They considered two different rescheduling time-domain determination methods during exploring the rescheduling implementation method and developed two problem-specific heuristics.

Pro-active scheduling focuses on improving the robustness of the initial scheduling solution by predicting the uncertain influence of the disturbance events. For example, Wang et al. [15] designed a knowledge-based evolutionary proactive scheduling approach and used a support vector regression to evaluate the robustness of the solution. Cui et al. [16] studied the integration of the production scheduling and maintenance planning in order to optimise the bi-objective of quality robustness and solution robustness for a flow shop with uncertain quality failure. Wu et al. [17] proposed a novel and comprehensive measure for schedule risk evaluation based on the internal relation among the total slack time, the probability and downtime of random machine breakdowns, and the makespan delay.

In all, it can be seen that reactive scheduling has good practicability in the manufacturing system with frequent dynamic events and high uncertainty. However, since the decision information is derived from local real-time information, it is difficult to predict the whole system's performance. Predictive-reactive scheduling is also suitable for online dynamic optimisation and is most widely used. It is a scheduling/rescheduling process that modifies the scheduling solution according to the dynamic events has good global scheduling performance. Pro-active scheduling has high robustness and can absorb dynamic disturbance to some extent. In order to improve robustness, pro-active scheduling can reduce machine utilisation. In summary, for the implicit disturbance caused by the deterioration effect and the explicit disturbance caused by the reprocessing of unqualified jobs considered simultaneously, the predictive-reactive scheduling strategy is adopted in this study.

However, most existed literature mainly optimise the efficiency of rescheduling solutions, which may result in the instability of scheduling schemes and the waste of many setup works. Therefore, it is necessary to consider both efficiency and stability in a predictive-reactive scheduling strategy [18]. Besides, the energy-saving and green development has been becoming the national development strategy in China, the study on scheduling needs to consider energy consumption as well as the traditional optimisation objectives such as the makespan, delivery time, and so on [19]. Inspired by this, the paper studies the DFJSP considering deterioration effect (DFJSP-DE) and energy consumption and the non-dominated sorting genetic algorithm III (NSGA-III) is designed to deal with both the implicit and explicit disturbance. To the best of our knowledge, this is the first study on the DFJSP-DE.

The main contributions of this study are as follows: (i) both the implicit disturbance and the explicit disturbance are considered in the DFJSP; (ii) the makespan, the energy consumption, and the stability are optimised simultaneously when generating a rescheduling solution with an improved NSGA-III. The rest of the paper is structured as follows. Section 2 introduces the DFJSP-DE. Section 3 formulates the DFJSP-DE. Section 4 proposes the NSGA-III for the DFJSP-DE. Section 5 reports the case study. Section 6 concludes the paper.

## 2 DFJSP-DE description

The DFJSP-DE can be described as follows. In those fine processing manufacturing system, there are \( n \) jobs to be processed, indexed by \( j = 1, 2, \ldots, n \) and \( m \) machines available, indexed by \( k = 1, 2, \ldots, m \). Operations follow a pre-defined precedence constraint. The operation \( O_j \) can be processed on any machine among a set of available machines \( M_j \subset \{1, 2, \ldots, m\} \) and occupies a different amount of the processing time on different machines. Besides, the actual processing time of the operation is variable with the influence of the deterioration effect. It consists of two parts: the fixed basic processing time and the variable penalty time [20]. Given a deterioration duration threshold, if the job cannot be processed before the given lower bound of the threshold, then the period of penalty time will be required. If the job is processed after the given upper bound of the threshold, the penalty time will be fixed. Within the threshold, the penalty time is the product of the deterioration rate and the starting time of \( O_j \), which means that the later the starting time is, the worse the deterioration effect will be.

The energy consumption per time unit of each machine varies with different speed levels. The unqualified jobs dynamically occur and need to be reprocessed immediately. The task for solving the DFJSP-DE is to assign a machine for each operation and sequence the operations on each machine in order to optimise the makespan, the energy consumption, and the stability of the rescheduling solution simultaneously.

## 3 Formulation for DFJSP-DE

### 3.1 Notations

Nomenclature lists all the notations.
3.2 Assumptions
To simplify the model, some assumptions are as follows.

(i) Jobs are independent and pre-emption is not allowed.
(ii) Each machine can process only one job at a time.
(iii) All jobs and machines are available at the beginning, i.e. \( t = 0 \).
(iv) Once a job finishes processing on a machine, it is immediately delivered to the next machine and the delivery time is negligible.
(v) The ideal processing time of all operations on their available machines is known.
(vi) The setup time is considered in the processing time and is negligible when a scheduling solution is generated.
(vii) The setup time is considered in the processing time and is negligible when a scheduling solution is generated.

3.3 Formulation of DFJSP-DE
Before the DFJSP-DE at the rescheduling point \( t_i \) is formulated, the energy consumption model and the deterioration effect model should be built. Both can be referred to as one of our previous studies [20]. At the rescheduling time point \( t_i \), the processed operations, the processing operations, the unprocessed operations, and the operations to be reprocessed should be marked first, and then the unprocessed operations and those in need of reprocessing will be rescheduled to optimise the makespan, the energy consumption, and the stability of the rescheduling solutions

\[
f = \min \left( C_{\text{max}}, E, W \right)
\]

\[
C_{\text{max}} = \max_{i,j} C_{ij}
\]

\[
E = \sum_{i=1}^{m} E^{(i)}
\]

\[
W = \sum_{j=1}^{n} \sum_{i=1}^{m} (\max \{(S_{ij}(t_i) - S_{ij}(t_{i-1})), 0\} + \gamma \max \{(S_{ij}(t_i) - S_{ij}(t_{i-1})), 0\} + \beta \times R_{ijk}(t_i)), \ k \in M_{ij}
\]

\[
p_{i\in,j} = p_{i\in,j} + b_k \left[ \max \left\{ t - t_{ij}^{\text{max}}, 0 \right\} - \max \left\{ t - t_{ij}^{\max}, 0 \right\} \right]
\]

\[
C_{ij}(t_i) - C_{ij}(t_{i-1}) \geq p_{i\in,j}X_{ijk}, \ i \in b_k(t_i), \ j \in a(t_i), k \in M_{ij}
\]

\[
C_{ij} - C_{ij-1} \geq p_{i\in,j}X_{ijk}, \ i \in n_j \text{ and } i \neq 1, k \in M_{ij}
\]

(see (8))

\[
\sum_{k=1}^{M_{ij}} X_{ijk} = 1, \ k \in M_{ij}, \forall i, j
\]

\[
p_{i\in,j} = p_{i\in,j} + p_{j\in,k} + p_{j\in,k}
\]

\[
E^{(i)} = E^{\text{off,off}} + E^{\text{cutting}} + F^{\text{cutting}} + E^{\text{idle}}
\]

\[
E^{\text{idle}} = \left[ \int_{S_{i1}}^{S_{i2}} P_{j}(t) \, dt + \int_{S_{i2}}^{S_{i1}} P_{j}(t) \, dt \right] \max \{X_{ijk}\}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

\[
E^{\text{cutting}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left\{ \int_{C_{ij} - C_{ij} - p_{ij}}^{C_{ij} - C_{ij} - p_{ij}} P_{j}(t) \, dt \right\} X_{ijk}
\]

where (1) indicates the optimisation objectives, the makespan \( C_{\text{max}} \), the total energy consumption \( E \), and the stability of the rescheduling solution \( W \) that should be optimised simultaneously. Equation (2) formulates the makespan objective. Equation (3) formulates the total energy consumption objective. Equation (4) formulates the stability objective. Equation (5) is the formula to calculate the actual processing time of \( O_{ijk} \). Equation (6) indicates the precedence relation. Equation (7) indicates the precedence relation of operations. Equation (8) indicates that one machine cannot process more than one job at a time. Equation (9) indicates that an operation can only be processed by one of the available machines. Equation (10) indicates that the processing duration of operation \( O_{ijk} \) consists of three parts: the duration for tool positioning and cutter's approach to job \( j \), the cutting duration, and the duration for cutter's departure from job \( j \). Equation (11) calculates the energy consumption of machine \( k \). Equations (12)–(15) are to calculate the energy consumption referred to Wu et al. [20]. Equations (16) and (17) define the range of decision variables.

4 Predictive–reactive rescheduling approach
4.1 Framework of predictive–reactive rescheduling approach
The framework of the predictive–reactive rescheduling approach is shown in Fig. 1. Before a dynamic event occurs, a predictive schedule considering the objectives of makespan and energy consumption is carried out. After a dynamic event occurs at the rescheduling time point, the NSGA-III is triggered to generate a set of non-dominated scheduling solutions by optimising the makespan, energy consumption, and stability simultaneously. The decision maker could select a reactive scheduling solution according to his/her preference. Then the selected reactive schedule will be executed until the next reprocessing event occurs.

4.2 NSGA-III for DFJSP-DE
4.2.1 Framework of NSGA-III: The NSGA-III is employed due to its outstanding performance [21]. The basic framework of the NSGA-III is similar to that of the non-dominated sorting genetic algorithm II (NSGA-II). The main difference between them lies in the selection mechanism. The details of the NSGA-III for DFJSP-DE are discussed as follows. Among them, please refer to Wu et al. [20] for the representation and decoding algorithm.

4.2.2 Crossover operator: Crossover operator is applied to generate offsprings. Among the various genetic algorithms for solving FJSP [22], the position-based crossover (PBX) and the
liner order crossover (LOX) are most frequently employed. Therefore, considering the representation characteristics of the chromosomes, the LOX and PBX are randomly selected to perform the crossover operations.

(i) The PBX operator
Step 1: Select some positions randomly and exchange the genes of the two parent individuals at the selected positions.
Step 2: Delete the same gene and insert in the remaining genes sequentially at the unselected gene positions (Fig. 2).

(ii) The LOX operator
Step 1: Select two positions randomly and exchange the fragments of the two parent individuals between the two positions.
Step 2: Delete the same gene and insert in the remaining genes sequentially outside the two intersections (Fig. 3).

4.2.3 Mutation operator: The two-point swap mutation operator is applied. The steps are as follows:
Step 1: Select two positions randomly.
Step 2: Swap the genes at the two positions (Fig. 4).

4.3 Rescheduling strategies
Three rescheduling strategies are employed, namely the full rescheduling, the local rescheduling, and the improved right shift rescheduling. When a dynamic event occurs, the operations to be reprocessed and the unprocessed operations need to be rescheduled, wherein the former has higher priority.

(i) The full rescheduling strategy: the full rescheduling strategy is to reschedule the operations to be reprocessed and unprocessed operations. Both the machine arrangement and processing sequence can be destroyed. The details are as follows.
Step 1: Generate a predictive schedule to optimise both the makespan and the energy consumption, and record the machine arrangement and the processing sequence of operations according to the predictive scheduling solution;
Step 2: Execute the predictive schedule until a reprocessing event occurs.
Step 3: Divide the tasks into the processed operations, the processing operations, the unprocessed operations, and the operations to be reprocessed.
Step 4: Reschedule the operations to be reprocessed and unprocessed operations to optimise the stability besides the makespan and the energy consumption, where the reprocessing operations are prioritised, return to step 2.

(ii) The local rescheduling strategy: The local rescheduling strategy is to keep the machines assigned to the unprocessed operations unchanged so that the stability can be optimised. The other steps are similar to those in the full rescheduling strategy.
(iii) The improved right shift rescheduling strategy: The improved right shift rescheduling strategy is to keep the machine arrangement and the processing sequence of the unprocessed operations unchanged and then reschedule the reprocessing and unprocessed operations. Adjust operation sequence according to the energy-saving scheduling heuristic considering the deterioration effect [20].

In summary, the three rescheduling strategies have their own advantages and disadvantages. The full rescheduling strategy has the largest solution space but spends long searching time. The solution space and searching time of the local rescheduling strategy are second, but the rescheduling solutions have high stability. The operation of the improved right shift rescheduling strategy is the easiest.

5 Case study

5.1 Design of experiments

5.1.1 Experiment setting: The proposed algorithm is coded with Matlab© and all the experiments are carried out on a desktop computer with Intel Core i5-3230, 2.60 GHz CPU, 2.00G RAM, and Win7 64 OS.

5.1.2 Data source: The instances from Brandimarte [23] are taken as the experiment data. Referring to Wang et al. [24], the deterioration rate of machine $k$ is generated randomly using the uniform distribution $U[0.005, 0.150]$. The values of the cutting power, the air cutting power, the idle power, and the turning-on/off energy consumption are set as those in [25], please see Table 1.

5.1.3 Parameters setting: The parameters are set as Table 2.

5.1.4 Aims: The aims of the experiments are as follows:

(i) to verify the effectiveness of the NSGA-III algorithm for solving multi-objective optimisation problems;
(ii) to compare the advantages and the disadvantages of three rescheduling strategies;
(iii) to analyse the rationality of the energy-saving scheduling algorithm considering the deterioration effect.

5.2 Results

5.2.1 Performance analysis of NSGA-III: In order to study the performance of NSGA-III for solving the many objectives scheduling problem, we test the DFJSP-DE. The NSGA-II is employed to compare with the NSGA-III. The parameters setting for the NSGA-II are referred to as Wu and Sun [25].

Test the Brandimarte [23] instances and run the NSGA-III and NSGA-II, respectively. The results are reported in Fig. 5–14. It can

![Fig. 5 Results of MK01](image)

![Fig. 6 Results of MK02](image)
be seen that for the instances MK01-MK04, there is no significant difference between the NSGA-III and the NSGA-II; for the instances MK05-MK10, the NSGA-III outperforms the NSGA-II significantly. In order to compare the NSGA-III and the NSGA-II intuitively, the Pareto solutions obtained with the NSGA-III and NSGA-II are first normalised and then summed. Equation (18) is for normalising and (19) is for summing

\[ f_{ji}' = \frac{f_{ji} - f_{j\min}}{f_{j\max} - f_{j\min}} \]  

\[ F = \sum_{j=1}^{3} \sum_{i=1}^{n} f_{ji}' \]  

where \( f_{j\min} \) represents the minimum value of the \( j \)th objective in the Pareto solutions obtained with the NSGA-II and NSGA-III, \( f_{j\max} \) represents the maximum value of the \( j \)th objective in the Pareto solutions obtained with the NSGA-II and NSGA-III, \( f_{ji} \) represents the value of the \( i \)th Pareto solution for the \( j \)th objective, \( f_{ji}' \) represents the normalised value of the \( i \)th Pareto solution for the \( j \)th objective, and \( F \) represents the sum of the normalised values.

The data are shown in Table 3 and presented in a line chart as shown in Fig. 15. It can be seen that for the instances, MK01–MK02, the NSGA-II outperforms the NSGA-III slightly; for the instances, MK03–MK10, the NSGA-III outperforms the NSGA-II.
significantly. Hence, for the many objectives of DFJSIP, the NSFAG-III has better performance than the NSFAG-II.

### 5.2.2 Comparison among three rescheduling strategies

Three rescheduling strategies are designed, i.e., the full rescheduling strategy, the local rescheduling strategy, and the improved right shift rescheduling strategy. In order to verify the effects of the three rescheduling strategies, MK07 is employed as the test instance. The rescheduling time point and rescheduling tasks are the same for the three rescheduling strategies. Run the algorithms ten times, respectively. The experimental results are shown in Figs. 16–25. It can be seen that, except for the fifth experimental result, the remaining nine results show the local rescheduling strategy performs best, followed by the full rescheduling strategy and the improved right shift rescheduling strategy.

In order to further analyse the performance of the three rescheduling strategies, the typical Pareto solutions for each strategy are selected for comparison. A typical Pareto solution from the Pareto set obtained with the full rescheduling strategy is selected with the stability priority, followed by makespan and energy consumption. With the selected typical Pareto solution as

| Instances | NSGA-III | NSGA-II |
|-----------|----------|---------|
| MK01      | 1.13     | 1.12    |
| MK02      | 1.31     | 1.29    |
| MK03      | 0.46     | 0.67    |
| MK04      | 0.98     | 1.09    |
| MK05      | 1.04     | 1.35    |
| MK06      | 1.05     | 1.45    |
| MK07      | 0.35     | 1.51    |
| MK08      | 0.67     | 0.97    |
| MK09      | 0.81     | 1.44    |
| MK10      | 0.90     | 1.25    |

Bold values mean that the NSGA-III performs better than the NSGA-II for the same instance.
the standard, a Pareto solution, which is better than the selected typical Pareto solution in terms of makespan, energy consumption, and stability is tried to be found from the Pareto set obtained with the local rescheduling strategy and the improved right shift rescheduling strategy. The objectives of the typical Pareto solutions for the three strategies are shown in Table 4 and are shown by line graphs, as shown in Figs. 26–28.

It can be seen from Table 4 and Figs. 26–28 that, for the stability objective, the results of ten rounds show the local rescheduling strategy performs best, followed by the full rescheduling strategy and the improved right shift rescheduling strategy. For the makespan objective, nine out of ten rounds of results show the local rescheduling strategy performs best, followed by the full rescheduling strategy and the improved right shift rescheduling strategy, and only in one round the full rescheduling strategy performs best, followed by the local rescheduling strategy and the improved right shift rescheduling strategy. For the energy consumption objective, in eight rounds the local rescheduling strategy performs best, followed by the full rescheduling strategy and the improved right shift rescheduling strategy, in one round the full rescheduling strategy performs best, followed by the local rescheduling strategy and the improved right shift rescheduling strategy, and in one round the local rescheduling strategy performs best, followed by the improved right shift rescheduling strategy and the full rescheduling strategy. Hence, for the DFJSP-DE, experiment results show the local rescheduling strategy performs best, followed by the full rescheduling strategy and the improved right shift rescheduling strategy. The Gantt chart for the initial scheduling solution is shown in Fig. 29. Also, the Gantt chart for a local rescheduling solution is shown in Fig. 30, where the coloured rectangles represent the reprocessing operation of the unqualified jobs.

5.2.3 Analysis of the influence of deterioration effect: The core of the approach for solving the DFJSP-DE is the energy-saving scheduling algorithm considering the deterioration effect. In order to study its performance, we carry out an experiment to compare the difference between the energy-saving scheduling algorithm considering the deterioration effect and that without considering the deterioration effect. Fig. 31 is the Pareto solutions for instance MK07 obtained with the two algorithms and it can be seen that the energy-saving scheduling algorithm considering the deterioration effect is superior to that without considering the deterioration effect. For a more intuitive representation, the experiment results obtained with the two scheduling algorithms are shown in box plots, as shown in Figs. 32–34. It can be seen from Figs. 32–34 that for all the three objectives, i.e. makespan, energy consumption, and stability, the scheduling results obtained with the energy-saving scheduling algorithm considering the deterioration effect are superior to those obtained with the energy-saving scheduling algorithm without considering the deterioration effect. Two typical Pareto solutions are selected from the Pareto sets obtained with the two decoding algorithms, respectively, according to the stability.
priority, followed by makespan and energy consumption. The
responding scheduling Gantt charts are shown in Figs. 35 and 36,
where the coloured rectangle represents the reprocessing operation
of unqualified jobs. On comparing Figs. 35 and 36, it is obvious

|   | Makespan | Energy consumption | Stability |
|---|----------|--------------------|-----------|
|   |          |                    |           |
| 1 | 298.80   | 295.90             | 329.76    | 2,558,459.08 | 2,675,398.87 | 2067.61 | 1813.53 | 2079.03 | 2561.36 | 2967.76 |
| 2 | 306.86   | 316.04             | 343.06    | 2,650,965.61 | 2,670,064.70 | 2,802,206.02 | 2631.14 | 2478.88 | 2552.95 | 2566.66 |
| 3 | 306.44   | 304.84             | 325.49    | 2,699,996.62 | 2,674,915.86 | 2,711,797.71 | 2648.11 | 2497.88 | 2570.60 | 2570.49 |
| 4 | 293.84   | 292.66             | 317.82    | 2,540,399.84 | 2,518,049.32 | 2,560,055.05 | 1999.14 | 1675.94 | 2293.72 |
| 5 | 302.99   | 302.68             | 324.68    | 2,597,492.05 | 2,597,201.40 | 2,641,038.74 | 2306.28 | 2079.03 | 2566.66 |
| 6 | 271.90   | 265.32             | 275.98    | 2,292,911.06 | 2,292,073.28 | 2,294,561.11 | 1090.55 | 878.08  | 980.81  |
| 7 | 306.13   | 303.13             | 326.31    | 2,628,705.25 | 2,608,613.38 | 2,660,563.52 | 2573.48 | 2207.89 | 2930.03 |
| 8 | 309.23   | 307.79             | 340.12    | 2,673,762.37 | 2,672,679.44 | 2,764,171.09 | 2601.17 | 2552.95 | 3266.66 |
| 9 | 321.72   | 296.50             | 322.24    | 2,659,305.33 | 2,576,812.28 | 2,637,161.79 | 2500.74 | 1990.60 | 2570.49 |
| 10| 311.04   | 309.36             | 332.43    | 2,699,409.28 | 2,674,398.55 | 2,743,431.33 | 2672.32 | 2487.77 | 3290.33 |

Table 4 Selected Pareto solution for three rescheduling strategies

Bold values represent the best result among the three rescheduling strategies for each test.
that the rescheduling solution obtained with the energy-saving scheduling algorithm considering the deterioration effect has a shorter makespan and less machine idle time.

Fig. 29 Gantt chart of the initial scheduling solution

Fig. 30 Gantt chart of the rescheduling solution with the local rescheduling strategy

Fig. 31 Comparison of the deterioration effect

Fig. 32 Box plot of the makespan comparison

Fig. 33 Box plot of the energy consumption comparison

Fig. 34 Box plot of the stability comparison

This study solves the DFJSP problems considering two types of disturbances, which are caused by the deterioration effect and the reprocessing of unqualified jobs and can easily destroy the stability of the scheduling solution. We take these two disturbances into...
account when modeling, and design an effective energy-saving decoding algorithm considering the deterioration effect according to the problem characteristics. Therefore, the stability of the rescheduling solution is improved obviously.

6 Conclusions

In this paper, we study the DFJSP-DE. An optimisation model is formulated to optimise the makespan, energy consumption, and stability simultaneously. The NSGA-III is designed to solve the problem. Three numerical experiments are carried out to verify the performance of the proposed algorithm. The main conclusions are as follows.

(i) For the many objectives optimisation problems, the performance of the NSGA-III is better than the NSGA-II.
(ii) For the DFJSP-DE, the results show that the local rescheduling strategy performs best, followed by the full rescheduling strategy and the improved right shift rescheduling strategy.
(iii) The deterioration effect is considered in the optimisation model, which reduces the impact of the deterioration effect on practical production.

To the best of our knowledge, this is the first study on the DFJSP-DE. However, there are still some limitations to our study. For example, only one kind of implicit disturbance and one kind of explicit disturbance are considered. In future research, it is necessary to further improve the deterioration effect model and the energy consumption model, and consider more kinds of disturbance events.

7 Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. 51305024.

8 References

[1] Pinedo, M.L.: ‘Scheduling: theory, algorithms, and systems’ (Springer, New York, 2008)
[2] Ouelhadj, D., Petrovic, S.: ‘A survey of dynamic scheduling in manufacturing systems’, J. Sched., 2009, 12, (4), pp. 417–431
[3] Sabuncuoglu, I., Goren, S.: ‘Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research’, Int. J. Comput. Integr. Manuf., 2009, 22, (2), pp. 138–157
[4] Vieira, G.E., Herrmann, J.W.: ‘Rescheduling manufacturing systems: a framework of strategies, policies, and methods’, J. Sched., 2003, 6, (1), pp. 39–62
[5] Lee, K., Zheng, F.F., Pinedo, M.L.: ‘Online scheduling of ordered flow shops’, Eur. J. Oper. Res., 2019, 272, (1), pp. 50–60
[6] Hopf, M., Thiel, C., Wendl, O.: ‘Competitive algorithms for multistage online scheduling’, Eur. J. Oper. Res., 2017, 267, (1), pp. 2007–2020
[7] Wang, Q., Tian, J., Fu, R.Y., et al.: ‘Online algorithms for scheduling on batch processing machines with interval graph compatibilities between jobs’, Theor. Comput. Sci., 2017, 700, pp. 37–44
[8] Lan, Y., Chen, X., Ding, N., et al.: ‘Online minimum makespan scheduling with a buffer’, Int. J. Found. Comput. Sci., 2014, 25, (5), pp. 525–536
[9] Wu, Q.D., Ma, Y.M., Li, L., et al.: ‘Data-driven dynamic scheduling method for semiconductor production line’, Contr. Theory Appl., 2015, 32, (9), pp. 1233–1239
[10] Shen, X.N., Yao, X.: ‘Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems’, Inf. Sci., 2015, 298, pp. 198–224
[11] Adibi, M.A., Zandieh, M., Amari, M.: ‘Multi-objective scheduling of dynamic job shop using variable neighborhood search’, Expert Syst. Appl., 2010, 37, (1), pp. 282–287
[12] Gao, K., Suganthan, P., Chua, T.: ‘A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion’, Expert Syst. Appl., 2015, 42, (21), pp. 7652–7663
[13] Kundukuri, N., Kulak, O.: ‘Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problems’, Comput. Ind. Eng., 2016, 96, (C), pp. 31–51

Fig. 35 Gantt chart of the rescheduling solution obtained with the decoding algorithm considering the deterioration effect

Fig. 36 Gantt chart of the rescheduling solution obtained with the decoding algorithm without considering the deterioration effect
[14] Liu, L., Zhou, H.: ‘Open shop rescheduling approach under unexpected arrival of a rush job’, Comput. Integr. Manuf. Syst., 2014, 20, (7), pp. 1631–1642

[15] Wang, D., Liu, F., Wang, Y., et al.: ‘A knowledge-based evolutionary proactive scheduling approach in the presence of machine breakdown and deterioration effect’, Knowledge-Based Syst., 2015, 90, pp. 70–80

[16] Cui, W.W., Lu, Z.Q., Li, C., et al.: ‘A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops’, Comput. Ind. Eng., 2018, 115, pp. 342–353

[17] Wu, Z., Sun, S., Xiao, S.: ‘Risk measure of job shop scheduling with random machine breakdowns’, Comput. Oper. Res., 2018, 99, pp. 1–12

[18] Gao, K., Yang, F., Zhou, M., et al.: ‘Flexible job shop rescheduling for new job insertion by using discrete Jaya algorithm’, IEEE Trans. Cybern., 2019, 49, (5), pp. 1944–1955

[19] Gao, K., Huang, Y., Sadollah, A., et al.: ‘A review of energy-efficient scheduling in intelligent production systems’, Complex Intell. Syst., 2019, doi: 10.1007/s40747-019-00122-6

[20] Wu, X.L., Shen, X.L., Li, C.B.: ‘The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously’, Comput. Ind. Eng., 2019, 135, pp. 1004–1024

[21] Deb, K., Jain, H.: ‘An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints’, IEEE Trans. Evol. Comput., 2014, 18, (4), pp. 577–601

[22] Wu, X.L., Wu, S.: ‘An elitist quantum-inspired evolutionary algorithm for the flexible job-shop scheduling problem’, J. Intell. Manuf., 2017, 28, (6), pp. 1441–1457

[23] Brandimarte, P.: ‘Routing and scheduling in a flexible job shop by tabu search’, Ann. Oper. Res., 1993, 41, (3), pp. 157–183

[24] Wang, T., Baldacci, R., Lim, A., et al.: ‘A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine’, Eur. J. Oper. Res., 2018, 271, (3), pp. 826–838

[25] Wu, X., Sun, Y.: ‘A green scheduling algorithm for flexible job shop with energy-saving measures’, J. Clean Prod., 2018, 172, pp. 3249–3264