Journal of Manufacturing Systems 65 (2022) 486–497

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Time-based resilience metric for smart manufacturing systems and optimization method with dual-strategy recovery

Qiang Feng [a], Xingshuo Hai [a,b], Meng Liu [a], Dezhen Yang [a,*], Zili Wang [a], Yi Ren [a], Bo Sun [a], Baoping Cai [c]

[a] School of Reliability and Systems Engineering, Beihang University, Beijing 100191, PR China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[c] College of Mechanical and Electronic Engineering, China University of Petroleum, Qiangdao, Shandong 266580, PR China

ARTICLE INFO

Keywords:
Smart manufacturing (SM)
Production scheduling
Recovery
Optimization
Resilience

ABSTRACT

Smart manufacturing (SM) enables production scheduling to automatically adjust the original plan to meet customer demands. The deep integration of advanced information technologies also makes SM systems prone to a wide range of possible attacks. However, rapid recovery in response to damage has not aroused enough recognition. This paper addresses the issue of resilience metrics and recovery optimization for SM systems. First, the characteristics of smart manufacturing scheduling (SMS), coupled with its execution, damage, and dual-strategy recovery behaviors, are analyzed. Afterward, a novel time-based resilience metric oriented toward the quantification of rapid recovery is proposed. Furthermore, a decision-making framework composed of a joint optimization model and a modified pigeon-inspired optimization algorithm with an enhanced learning strategy and crossover operator (PIOLC) is established. Last, a case study of a flexible job shop scheduling problem with 8 jobs and 8 machines is conducted to verify the effectiveness of the work. Experimental results show that the proposed approach can achieve dual-strategy recovery optimization by increasing the system resilience to 86.2%.

1. Introduction

Developments in the integration of information technology, computing capacity, and artificial intelligence accelerate the digital transformation of manufacturing from a primarily physical process to smart manufacturing (SM) [1,2]. Production scheduling is an important operational activity with regard to the allocation of service resources that is required to minimize human involvement to meet customer demands in a SM environment [3]. The characteristics of smart manufacturing scheduling (SMS) on levels of technical sophistication, integration and automation magnify its advantages of being efficient and autonomous, while also increasing the risk of being attacked, whether over network links or physical devices [4]. The threats facing large-scale deployment SMS systems are critical and with more incidents than equipment failures and production losses. The consequences extend well beyond harm to the industrial process.

To response to damage and ensure on-time delivery after disruptions, efficient recovery policies need to be automatically generated in an acceptable timeframe to meet the requirements of practical production. Hence, to recover from disruptions and sustain production, a systematic scheme accompanied by: 1) an accurate performance measure, 2) a feasible decision-making framework, and 3) an effective optimization approach, is indispensable. This study investigates the problem of resilience metrics for SMS and its recovery optimization.

Resilience is widely used to describe the capacity to adapt and recover from disruptions and has attracted attention in manufacturing systems in recent years. To this end, a number of authors have proposed research into investigations of how resilience can be developed to reduce the impact of threats [5–8]. In spite of this, the aforementioned studies pay more attention to resilient control methodologies rather than the design of a proper resilience metric. Although these indicators are basically obtained by performance losses or attribute-based methods [9], they are not necessarily applicable to a SM system. By introducing “Resilient Operator 5.0″, the need for resilience was emphasized in [10], especially for SM systems, it would be upgraded to an agile and reconfigurable pattern where advanced technologies were used to react and

* Corresponding author.
E-mail addresses: fengqiang@buaa.edu.cn (Q. Feng), haixingshuo@buaa.edu.cn (X. Hai), ZY2114124@buaa.edu.cn (M. Liu), dezhenyang@buaa.edu.cn (D. Yang), wzl@buaa.edu.cn (Z. Wang), renyi@buaa.edu.cn (Y. Ren), sunbo@buaa.edu.cn (B. Sun), caibaoping@upc.edu.cn (B. Cai).

https://doi.org/10.1016/j.jmsy.2022.08.010
Received 16 May 2022; Received in revised form 30 July 2022; Accepted 29 August 2022
Available online 20 October 2022
0278-6125/

recover from a disruption. Though delivering increased resilienc, there is presently a lack of empirical evidence showing indications of how it can be built in the digital era. Recent perspectives have concentrated on a quick recovery [11] and rapid changes [12] in existing organizational processes, which can be understood as positively adjusting to shocks as well as responding to the demand for on-time delivery. Accordingly, the resilience indices related to time have been put forward to assist systems to recover faster from a disruptive event. The recovery speed was considered as a metric in [13] where the performance in resilience measure was quantified based on the ability to meet end-user demands. In an attempt to be more focused, Carvalho *et al.* [14] introduced a resilience index of on-time delivery to two specific failure modes. Nevertheless, a simple aggregation method was unable to guarantee the independence between performance and time variables. Besides, an empirical-based approach with a special case showed a lack of generality. Before that, Henry *et al.* [15] proposed a generic resilience metric that was quantified as a function of time, whereas the rapid recovery was not easy to embody in such cases. According to Romero *et al.* [16], the build-up of manufacturing resilience required innovative combinations of adapting existing production capabilities and developing new ones by reframing current patterns. To the best of author's knowledge, no study by far has proposed a unified metric considering both fully time-based variables and efficient practice that can quantify smart manufacturing resilience of rapid recovery. To fill the gap, we attempt to exploit the resilience metric and corresponding recovery strategies in the direction of production innovation based on the conceptual framework of manufacturing resilience engineering.

Popular recovery strategies in response to disruptions, rescheduling and maintenance have attracted many researchers to assign manufacturing scheduling. The former minimizes the adverse effect by updating an existing schedule and planning activities for the next time period [17], while the latter is carried out to respond to damage and rectify faults [18]. For this reason, both rescheduling and maintenance deal with not only the response to disruptions [19,20], but also the optimization of performances in production [21,22]. However, in most of the literature, these two items have been employed separately, which limits the acquisition of high-quality solutions, particularly under inevitable dynamic events [23].

To balance trade-offs between the two parts, the integrated optimization of maintenance and scheduling has received more attention in recent years. To minimize the expected make-span, Hu *et al.* [24] proposed a joint decision-making strategy combining job scheduling and preventive maintenance for a two-machine flow shop problem. Similarly, Yang *et al.* [25] formulated the integrated optimization problem as Markov decision process where a learning-based algorithm was used to find the optimal production policy. By integrating a flexible job shop problem into a mixed-integer programming model, Ghaleb *et al.* [26] considered the real-time joint optimization of maintenance and scheduling to address a series of uncertainties in a SM system. To fully consider production efficiency and machine reliability, Chen *et al.* [27] established an integrated multiobjective optimization model with flexible scheduling and accurate maintenance. According to the conducted literature, the addressed criteria mainly concern with make-span, costs, tardiness and reliability of machines. Additionally, there is a lack of a unified decision-making framework on the integration of rescheduling with maintenance by addressing the issue of resilience-oriented recovery optimization. In fact, when regarding rescheduling as an automatic process while performing maintenance by human operators, this integration will become a necessity to build SM resilience which can be explained based on the data analysis in [28]. Specifically, the cooperation ensures the optimal operation of a SM system [10]. Consequently, formulating SM resilience into a unified dual-strategy optimization model by simultaneously considering rescheduling and maintenance policies deserves more attention. The focus in this paper is on extending the current work by developing a new scheme in this area.

As a bioinspired swarm intelligence algorithm, pigeon-inspired optimization (PIO) and its variants were proven competitive in the quick speed of convergence and superior optimal solutions[29–31]. Unlike other heuristic approaches widely implemented for rescheduling and maintenance optimization problems, little work has been conducted in manufacturing using this new optimizer. While Wu et *al.* [32] and Fu *et al.* [33] extended PIO to multiobjective cases and developed hybrid methods to solve production scheduling problems, the objectives were similar to other research studies. In addition, the lack of population diversity of the original algorithm limits its potential in complex problems [34]. Thus, in this research, a PIO algorithm that introduces an enhanced learning strategy and crossover operator is utilized to achieve resilience-oriented dual-strategy optimization.

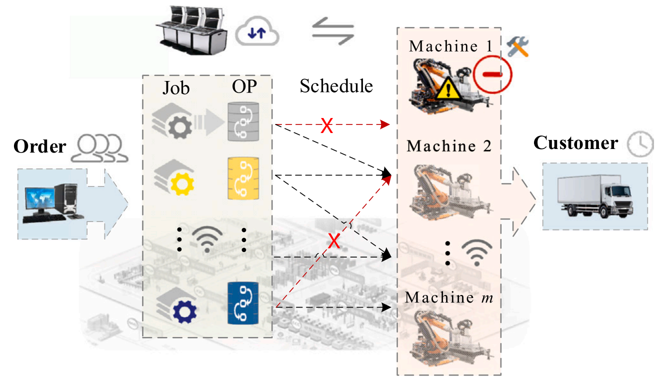

Fig. 1. A conceptual diagram of the smart manufacturing job shop scheduling procedure.

Despite numerous research studies, it is still challenging to elaborate unified and integrated approaches able to quantify and optimize the resilience of a SMS to identify the optimal decision-making. Thus, this research is devoted to providing an overall solution to address the above issues. The unique contributions of this paper include: 1) A novel time-based resilience metric is given for SMS systems. By utilizing the resilience triangle as the underlying rationale, this proposed resilience metric considers both fully time-based variables and efficient strategies that can quantify smart manufacturing resilience of rapid recovery; 2) A solution framework is established based on a joint optimization model considering maintenance and rescheduling strategies simultaneously. As a unified decision-making scheme, it is applicable to help management make well-informed decisions to fulfil production demand; 3) A new variant of the PIO algorithm with an enhanced learning strategy and a crossover operator is proposed where the learning strategy, a parameter updating mechanism and population diversity are promoted to support the resilience optimization with dual-strategy recovery.

The remainder of this paper is organized as follows: Section 2 is devoted to a description of the problem. The characteristics of SMS, attack-oriented recovery policies and a graph-based dual-strategy representation are detailed in this section. Section 3 introduces the time-based resilience metric. The solution framework is established in Section 4 where a joint optimization model and the proposed PIOLC algorithm are given. A case study for a flexible job shop scheduling problem is presented in Section 5. Finally, Section 6 concludes the paper and gives suggestions for future research.

2. Problem descriptions

2.1. Characteristics of scheduling in the SM environment

In the SM environment, the scheduling problem is generally considered a system with both self-management capabilities and the mechanism of human-technology integration [35,36]. As a typical task of SMS, job-shop scheduling considers a set of n jobs $\{J=J_1,J_2,\ldots,J_n\}$ are processed on a set of m machines $\{M=M_1,M_2,\ldots,M_m\}$ at the beginning of the schedule. During the production process, each job J_i, i = 1,2,…,n contains a predetermined sequence of operations $\{O_{i,1},O_{i,2},…,O_{i,ki}\}$, where k_i represents the total number of operations of J_i. Note that each operation $O_{i,j}$, j = 1,2,…,k_i can be executed by a set of candidate machines $M(O_{i,j})\subseteq M$, with the processing time of the same operation varying on different machines, which is denoted by p_{ijk}.

On this basis, a conceptual diagram (see Fig. 1.) of the procedure is given with the main characteristics:

1) The workshop is characterized by large-scale, distributed deployment;
2) The communication and data transmission between devices are realized through high-speed mobile internet;
3) The functions of the control center include receiving orders, online monitoring, real-time decision-making, and service assessment;
4) There is always an interaction of humans and technology in the scheduling process.

2.2. Recovery policies

During the whole production scheduling period, random attacks can inevitably occur and cause damage to the machine. To cope with disruptions, two widely used recovery policies are investigated where random attacks arrive following a nonhomogeneous Poisson process [37].

2.2.1. Rescheduling strategy

In response to changes during production, rescheduling generates a new intersected schedule that changes the scheme of initial operations. In this work, a total rescheduling method, called regeneration [17], is adopted that reschedules the entire set of operations not processed whether they were affected or not. Accordingly, it is a concrete policy produced by a designed algorithm to determine how the remaining operations are rearranged. Thus, a rescheduling strategy involves 1) updating an existing schedule and 2) generating a new schedule that consists of assigning both a set of machines $M(O_{i,j})$ and a start time $t^s_{(i,j)}$ to each unfinished operation in the set $O=\cup_{1\leq i\leq n}O_i$. Note that decision-making is instantaneous, and a rescheduling point is a disruptive time.

2.2.2. Maintenance strategy

As another recovery strategy to cope with disruptions, the maintenance strategy is applied to restore the system to an as-good-as-new state by directly repairing damaged machines or replacing failed components [38]. Despite various types of maintenance policies, not all of them are suitable for handling emergencies since unintended events will inevitably take place. As emphasized in [10], one of the stops on the road to failure of any system is failing to anticipate a problem before it has arrived. Additionally, considering the need of low-cost and time-saving, a widely used corrective maintenance (CM) strategy is adopted in this work.

By describing a SM system as the marriage of information, technology, and human ingenuity [10], human operators are capable of finding solutions to problems without obvious tools and provide the highest contribution to the system resilience when adapting to an unexpected change. With that in mind, the specific implementations of a CM strategy, involving repairing a faulty machine or replacing a failure component with another new one, are achieved by human operators. As a result, the interrupted machine(s) resumes its initial state, after performing CM for a period of time. It should be noted that the decision-making time is negligible; hence, the strategy can be expressed by the average maintenance time.

Suppose an attack occurs at time t_d, causing damage to m machines denoted by $M_1,M_2,…,M_m$, respectively. For any machine M_k, k = 1,2,…, m, the average maintenance time required to perform a maintenance strategy AT_k is a constant value depending only on the machine itself.

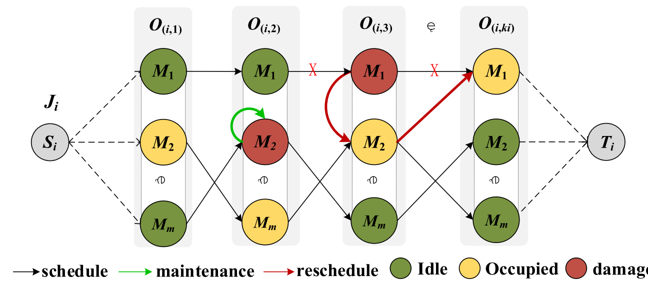

Fig. 2. Directed graph representing feasible solutions of each job.

2.3. A graph-based dual-strategy representation

Despite the popularity of rescheduling and maintenance in dealing with disruptions, the two strategies are independently carried out during a manufacturing process; however, there are two sides to their implementation due to the uncertainty of attacks and the degree of product completion.

In this study, rescheduling and maintenance strategies are integrated into a directed graph (see Fig. 2, where the vertices in the same column correspond to the alternative machines of an operation, while the arc of the graph represents the schedule of an operation. As shown in Fig. 2, a feasible solution of any job J_i, i = 1,2,…,n processing on different machines can be viewed as a full path from the dummy start node S_i to the dummy end node E_i. In the meantime, each machine M_k has three status including “idle”, “unavailable” and “occupied” that can be converted to each other, where the value S_k=0 stands for an idle state (no operation processed, colored in green), S_k= −1 represents an unavailable state (machine breakdown, colored in red), and S_k= 1 represents an occupied state (in operation, colored in yellow). Only when the machine is idle can an operation be processed on it; otherwise, it will either wait or be replaced with a new route.

It should be noted that the processing routes of operations through the machines are dynamically generated during the production phase, with the aim being, to find an optimal order such that the objectives are optimized. In particular, with disruptions incorporated, production scheduling requires that optimize rescheduling and maintenance jointly, in practice, to obtain satisfactory solutions. Thus, according to the definition in Section 2.2 and the graph-based method, the rescheduling strategy can be viewed as choosing another available node to relink the path (the arc colored in red), while maintenance is expressed by waiting in place (the arc colored in green) until the status changes from S_k= −1 to S_k= 0.

3. Time-based resilience metric

In this section, the proposed resilience metric based on completion

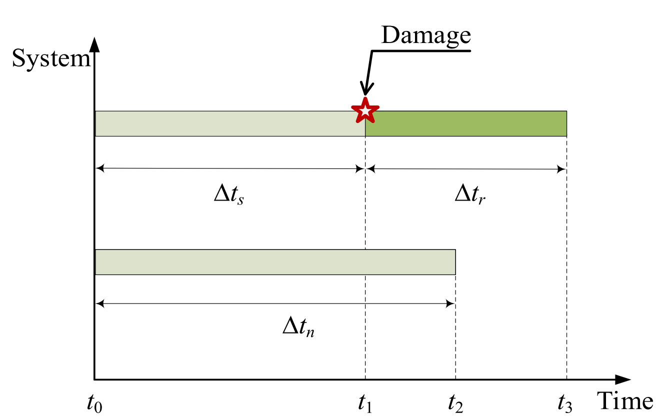

Fig. 3. Time axis of a scheduling system with disruptions.

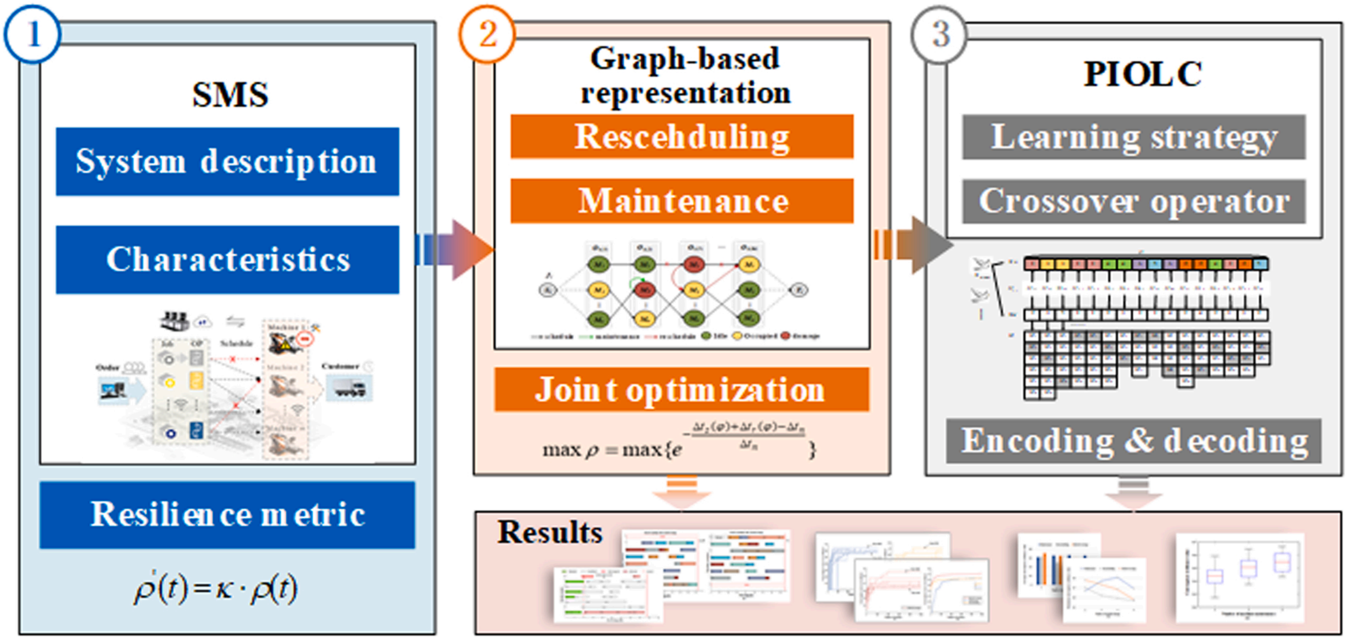

Fig. 4. Solution framework of the proposed method.

time for a SMS system is introduced. Then, the supplementary element needed for its computation is detailed, thereby developing a thorough and inclusive system of resilience.

3.1. Resilience metric based on completion time

Resilience is considered an inherent attribute of an engineering system covering time-related properties that is determined by maintenance resources [39,40]. The required resources are sufficiently provided with the continuous improvement of productivity under the background of SM. Therefore, the time factor itself becomes one of the critical performance criteria for a scheduling system whether disruptions occur or not.

Considering the time of a scheduling procedure as illustrated in Fig. 3, the task starts from the initial time t_0 to the completion time t_2 of the last finished operation. The duration of the process without disruptions $t_2 - t_0$ is denoted by Δt_n which is also recognized as a common criterion "make-span" in scheduling problems. Suppose a disruptive event occurs at time $t_1 (t_1 < t_2)$, resulting in machine breakdowns and the interruption of ongoing operations. Recovery actions, including maintenance and rescheduling, will be carried out instantaneously until all remaining operations are completed at t_3. Thus, the total completion time is extended to $t_3 - t_0$, in which $\Delta t_s = t_1 - t_0$ and $\Delta t_r = t_3 - t_1$ indicate the execution time before and after the disruption, respectively.

According to previous descriptions, the ability of the system to respond to failures is related to the duration of recovery strategies Δt_r. However, the external disruptions should not be involved in the quantification; thus, a ratio of delay $\frac{\Delta t_s+\Delta t_r-\Delta t_n}{\Delta t_n}$ is used as a substitute to precisely eliminate randomness. On this basis, the time-based resilience $\rho(t)$ proposed in this study can be defined as "the efficiency of accurately completing the schedule of a set of jobs" via the following equation:

$$\rho(t) = e^{-\frac{\Delta t_s+\Delta t_r-\Delta t_n}{\Delta t_n}} \tag{1}$$

where ρ is a real number within (0,1] and the natural exponential function $e^{(.)}$ is used to balance the change effect of ρ and make it easy to integrate. It is obvious that the value of resilience ρ increases, with decreasing recovery duration.

3.2. Integrity-based correction factor

Based on the previous definition, the proposed resilience metric is a completely time-based function, which benefits from the viability and sustainability of production in SM environment. However, the rise of the Industrial 4.0 has always brought about increased risks [41], such as a pandemic, severe demographic challenges, local war, etc. To further perfect the metric and provide a more comprehensive quantification, an integrity coefficient κ, $\kappa \in (0,1]$ is detailed in this part. It should be noted that the integrity coefficient is used simply as a correction factor and is not always incorporated in the formulation. For example, when resources are limited, it can be regarded as a factor to ensure the completion of scheduling.

The integrity coefficient reflects the degree of the resources available in a scheduling process. In this proposed formula, it is also employed as a correction factor of resilience and is quantified by the changes in system resources before and after adopting recovery strategies, which can be mathematically shown as follows:

$$\kappa = e^{-\frac{\sum k_i n_i - \sum k_i n_i'}{\sum k_i n_i}} \tag{2}$$

where $n_i \in R$ denotes the available quantity of scheduling products of type $i \in N^*$ under normal conditions, and k_i denotes the corresponding importance correction factors. On this basis, $n_i^{'}$ and $k_i^{'}$ represent the available machines and corresponding importance correction factor, respectively, at the end of recovery.

Let $\rho'(t)$ be the resilience of a scheduling system at time t. In its basic form, $\rho'(t)$ describes the efficiency of the scheduling completion, as considered to be the product of two components in the following equation:

$$\rho'(t) = \kappa \bullet \rho(t) \tag{3}$$

Evidently, the value of resilience ρ' is within (0,1] and increases with the increase in ρ when κ remains constant.

4. Solution framework and method

To address the issues previously described, this section introduces the modeling of the joint optimization problem, the solution framework, and the proposed approach. The framework incorporates maintenance with rescheduling by merging and simplification of node status, which allows a more flexible and efficient dual-strategy method compared with any single strategy. As a result, the enhanced PIO algorithm can be implemented on the joint optimization model.

4.1. Solution framework

The purpose of the optimization is to produce a sequence of operations to be processed on each machine with their processing start times in such a way that the given resilience metric is maximized and the problem constraints are satisfied. According to the definitions in Section 3.1, the proposed resilience is only related to the recovery policies; thus, Eq. (1) can be further written as:

$$\rho(\varphi) = e^{-\frac{\Delta t_s(\varphi)+\Delta t_r(\varphi)-\Delta t_n}{\Delta t_n}} \tag{4}$$

where φ consists of rescheduling φ_{res} and maintenance φ_{rep} which is expressed by $\varphi = \varphi_{rep} \cup \varphi_{res}$.

Based on the graph-based representation method, we can further analyze the two strategies from the system level, that is, once the maintenance is conducted, there is no difference between the current unavailable status and the occupied status of a machine. To simplify the problem, maintenance can be regarded as a special occupancy state; thus, there will be two states of the machine, in which $S_k = 1$ indicates an occupancy, $S_k = 0$ denotes an idle state. The rescheduling strategy involves determining the machine-operation allocation, as well as the scheduled start time of each remaining unprocessed or unfinished operation. The maintenance process can be considered as a special kind of operation with a certain period of processing time MT_k.

In order to further clarify the solution of dual-strategy recovery in this work, a framework can be established, which is shown in Fig. 4.

4.2. Modeling of the joint optimization problem

4.2.1. Notations for variables

When the system is set for maintenance at the end of the current mission, a component can be either in a functioning state or in a failed state. Hence, two decision variables are used to describe the status of an operation $O_{i,j}$ at the beginning of a scheduling and a rescheduling process:

$$x_{(i,j,k)} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{5}$$

$$y_{(i,j,k^*)} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{6}$$

where $x_{(i,j,k)}$ and $y_{(i,j,k^*)}$ are both binary and determine whether machine $M_k(M_{k^*})$ processes operation $O_{i,j}$, where 1 means processed and 0 means unprocessed.

4.2.2. Objective function and constraints

Based on the recovery of a scheduling system and the objective functions, the optimization model is as follows:

$$\max\rho = \max\{e^{-\frac{\Delta t_s(\varphi)+\Delta t_r(\varphi)-\Delta t_n}{\Delta t_n}}\} \tag{7}$$

subject to the following constraints:

1) Only one machine can be selected for each operation at a time:

$$\sum_{k=1}^{m} x_{(i,j,k)} = 1, \forall i,j \tag{8}$$

$$\sum_{k^*=1}^{m} y_{(i,j,k^*)} = 1, \forall i,j \tag{9}$$

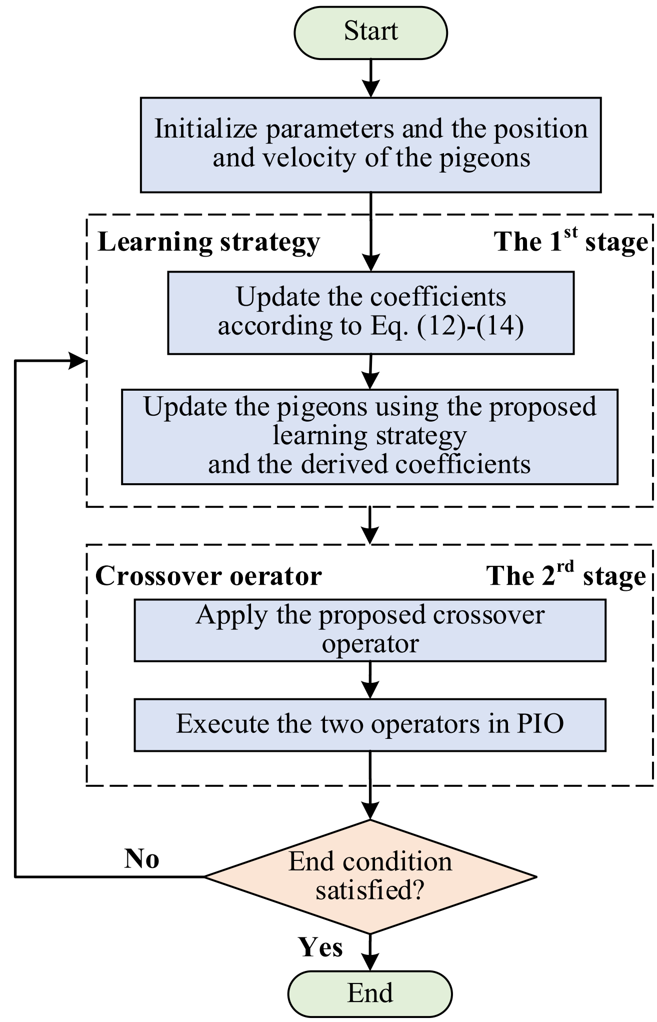

Fig. 5. Flowchart of PIOLC.

where $x_{(i,j,k)} + y_{(i,j,k^*)} = 1$ should be satisfied.

2) The processing duration of $O_{i,j}$, on machine M_k, denoted by $pd_{(i,j,k)}$, is determined by its start time $t^s_{(i,j,k)}$ and completion time $t^c_{(i,j,k)}$:

$$t^s_{(i,j,k)} + x_{(i,j,k)} pd_{(i,j,k)} = t^c_{(i,j,k)} \tag{10}$$

3) The time constraints between the two adjacent operations ($O_{i,j}$ and $O_{i,j+1}$) of the same job:

$$\begin{Bmatrix} t^c_{(i,j,k)} - t^c_{(i,j,k^*)} - pd_{(i,j+1,k)} \geq 0 \\ t^s_{(i,j,k)} - t^c_{(i,j,k^*)} \geq 0 \end{Bmatrix}, k \neq k^*, j \leq j+1 \tag{11}$$

4.3. Enhanced pigeon-inspired optimization

Despite the promising results in different aspects of practical issues, PIO still suffers from premature convergence of local optima. Considering the limitations of the algorithm mainly depend on the coefficient and learning mechanism [34], we introduce a hybrid strategy to strengthen the performance of the algorithm and apply it to the joint optimization problem.

4.3.1. Flowchart

In the basic PIO algorithm, each pigeon i ($1 \leq i \leq N$) has the position vector X_i^{iter} and the velocity vector V_i^{iter}, which are updated following the map and compass operator and the landmark operator [29] in each

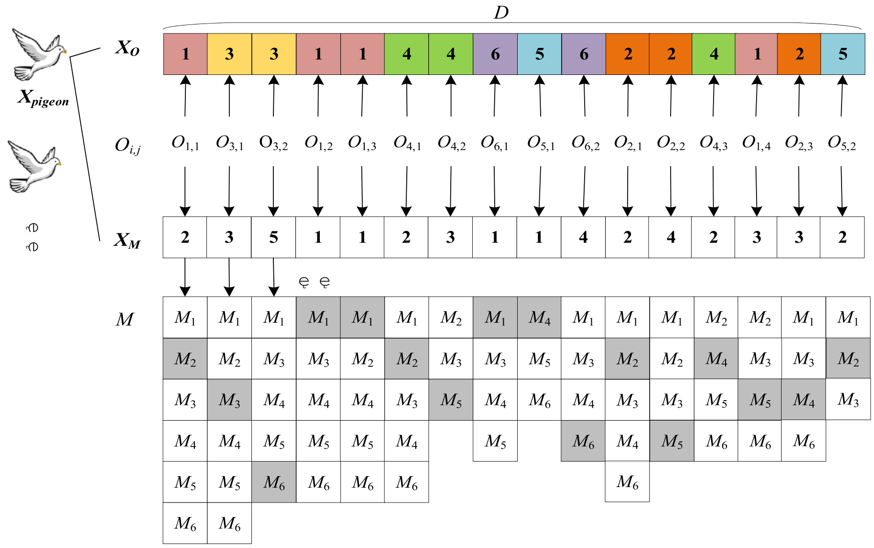

Fig. 6. Encoding scheme of a pigeon.

iteration *iter*. In the first stage, the pigeons fly through the *D*-dimensional solution space under the guidance of their own global best position X_{gbest}, while the second stage is implemented depending on the central position of the selected individuals X_{center}. In this work, the improvements are summarized as follows:

1) The learning strategy is enhanced to make full use of the knowledge of all pigeons to regulate the motion scheme;
2) A reasonable parameter updating mechanism is introduced to improve the flexibility of the algorithm for various problems;
3) A new arithmetic crossover operator is utilized to increase the population diversity during the search process.

On this basis, the flowchart of the PIOLC algorithm can be illustrated in Fig. 5.

4.3.2. The learning strategy and parameter updating mechanism

Based on the analysis in [34], the proper learning strength from the global best position X_{gbest} and the center X_{center} is important to improve algorithm performance. Accordingly, in our proposed scheme, each pigeon takes advantage of the experiences of the population, i.e., the global best position and the center of all pigeons contribute to V_i^{iter} and X_i^{iter}. Thus, the social learning components of V_i^{iter} and X_i^{iter} comprise terms of form ($X_{gbest}- X_i^{iter}$) and ($X_{center}- X_i^{iter}$). The important concern in this regard is to define proper coefficients for the mentioned terms.

In the first stage, the map and compass operator provide navigation cues for each pigeon. The more capable pigeons play more important roles in the optimization. Therefore, the coefficient is defined to be proportional to the fitness of the global best position of the pigeons. To avoid a local optimum, randomness is included in our design. Moreover, a separate social learning coefficient for each pigeon is also given to improve the search ability. From the foregoing, the social learning coefficient of the map and compass operator $M^{iter}= [m_1^{iter},m_2^{iter},\ldots,m_N^{iter}]$ is computed as:

$$M^{iter} = F^{iter} \bullet Rand^{iter} \tag{12}$$

where $F^{iter}= [f_1^{iter},f_2^{iter},\ldots,f_N^{iter}]$ denotes the normalized fitness values of the pigeons and *Rand* represents a normalized nonnegative random matrix of dimensions $N \times N$ whose elements r_{ji}^{iter} are chosen within the range of [0,1]. More precisely, f_i^{iter} is calculated as:

$$f_i^{iter} = \frac{X_i^{iter} \bullet fit(X_i^{iter})}{\sum_{j=1}^{N} X_j^{iter} \bullet fit(X_j^{iter})} \tag{13}$$

where $fit(\cdot)$ represents the fitness of a given position. Using Eq. (12), $m_i^{iter}\in[0,1]$ is derived as:

$$m_i^{iter} = f_1^{iter} \bullet r_{1i}^{iter} + \cdots + f_N^{iter} \bullet r_{Ni}^{iter} = \sum_{j=1}^{N} f_j^{iter} \bullet r_{ji}^{iter} \tag{14}$$

In the second stage, pigeons unfamiliar with landmarks are abandoned, and the remaining pigeons dominate the direction of the algorithm. Compared with the global best position X_{gbest}, the center X_{center} has a stronger effect on escaping from a local optimum. As a result, the coefficient of the term ($X_{center}- X_i^{iter}$) should be greater than the component in the first stage, which is defined as:

$$l^{iter} = \max\{m_i^{iter}\} \tag{15}$$

where l^{iter} will be within the range of [0,1].

The last parameter, representing individual inertia, is the map and compass factor *R*, which is set to a constant value in PIO and most of its variants. In this work, *R* is decreased over iterations using the following equation:

$$R^{iter+1} = \omega \bullet R^{iter} \tag{16}$$

where ω is a damping factor that is set to 0.87 and the initial value of *R* is set to 1.

4.3.3. The crossover operator

To increase the diversity of the population, the crossover operator is conducted between a selected pigeon p_s and a random pigeon p_r with probability P_c^{iter}. In our design, the offspring is created by:

$$X_i^{iter} = X_i^{iter} + \mu^{iter} \bullet X_r \tag{17}$$

where μ^{iter} denotes the weight parameter and X_r is the position of p_r. On this basis, the offspring substitutes for the original pigeon.

Clearly, the diversity of the population can be increased by using the crossover operator. However, for a production scheduling optimization problem, the convergence speed and accuracy should be more considered. Thus, to further facilitate the convergence performance, two damping parameters λ_P and λ_μ are applied to P_c^{iter} and μ, respectively, which can be computed as

Table 1
Processing time of operations (minutes).

Job	Operation	Processing duration (min)							
		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
J1	*O1,1*	5	6	9	4	7	8	7	–
	O1,2	7	–	9	6	5	10	–	6
	O1,3	4	5	–	7	6	3	5	–
	O1,4	–	2	3	–	5	4	8	6
J2	*O2,1*	5	6	8	9	–	10	7	–
	O1,1								
	O1,1								
	O2,2	9	8	7	–	4	–	7	5
	O2,3	3	–	6	8	–	2	4	7
	O2,4	10	8	9	6	4	7	–	–
J3	*O3,1*	5	8	9	11	6	7	4	–
	O1,1								
	O1,1								
	O3,2	9	–	8	6	12	15	10	7
	O3,3	2	4	5	6	–	10	–	–
	O3,4	9	9	–	8	5	6	7	3
J4	*O4,1*	11	12	9	14	–	10	8	–
	O1,1								
	O4,2	7	–	5	–	6	–	9	–
	O4,3	–	10	–	7	8	11	9	–
	O4,4	5	8	5	9	–	7	–	10
J5	*O5,1*	5	6	7	8	9	–	10	–
	O1,1								
	O5,2	10	–	7	4	9	8	6	–
	O5,3	–	9	8	7	4	5	7	–
	O5,4	9	9	–	6	7	5	4	6
J6	*O6,1*	6	7	5	4	6	9	–	3
	O1,1								
	O1,1								
	O6,2	11	–	9	9	9	7	6	4
	O6,3	10	8	9	10	11	–	10	–
	O6,4	–	–	9	10	11	8	9	10
J7	*O7,1*	5	4	7	6	7	–	10	–
	O7,2	–	9	–	–	11	7	–	–
	O7,3	–	8	9	3	8	6	–	10
	O7,4	7	8	9	3	8	6	10	2
J8	*O8,1*	2	6	5	9	–	4	–	7
	O8,2	7	4	7	8	9	–	10	–
	O8,3	5	5	–	8	5	6	7	4
	O8,4	3	5	–	–	8	7	–	–

Processing time of operations (minute).

$$P_c^{iter+1} = \lambda_P \bullet P_c^{iter} \tag{18}$$

$$\mu^{iter+1} = \lambda_\mu \bullet \mu^{iter} \tag{19}$$

where the initial values of λ_P and $\lambda_\mu$are set to 1.

4.3.4. Encoding and decoding mechanism

In our proposed algorithm, two vectors are included in each pigeon: the operation scheduling vector and machine assignment vector, which are represented by X_O and X_M, respectively. The dimension D of each vector is equal to the number of all operations in the scheduling, rescheduling jobs. As shown in Fig. 6, the encoding scheme is given by an operation-based representation method. In the row of operation scheduling vectors, each number corresponds to a job and is distinguished by various colors. In the row of machine assignment vectors, the number represents the index of the machine in an available machine set M of each operation. Details of this example with 8 jobs and 8 machines are listed in Table 1. Thus, the operation scheduling vector X_O= [1, 3, 3, 1, 1, 4, 4, 6, 5, 6, 2, 2, 4, 1, 2, 5], where the first element 1 denotes the first operation of J_1, and the last element 5, which appears twice, indicates the second operation of J_5. The machine assignment vector X_M= [2, 3, 5, 1, 1, 2, 3, 1, 1, 4, 2, 4, 2, 3, 4, 2], where the third element 5 indicates that $O_{3,2}$ is processed on a machine with an index value of 5 from the available machine set $\{M_1,M_3,M_4,M_5,M_6\}$, which is denoted by M_6.

Since decoding can be interpreted as the reverse of encoding, it is executed by returning the two vectors back into the solutions with considerations of constraints (1), (2) and (3).

5. Case study

To evaluate the effectiveness of the proposed method and the performance of the PIOLC algorithm, an example of a flexible job shop scheduling problem with 8 jobs and 8 machines is investigated as the case study.

5.1. Implementation and parameter settings

The PIOLC is written and implemented in MATLAB R2018b on an Intel (R) Core (TM) i5–6200 personal computer running on a 2.30 GHz processor with 8 G RAM memory. The detailed processing time of operations is listed in Table 1.

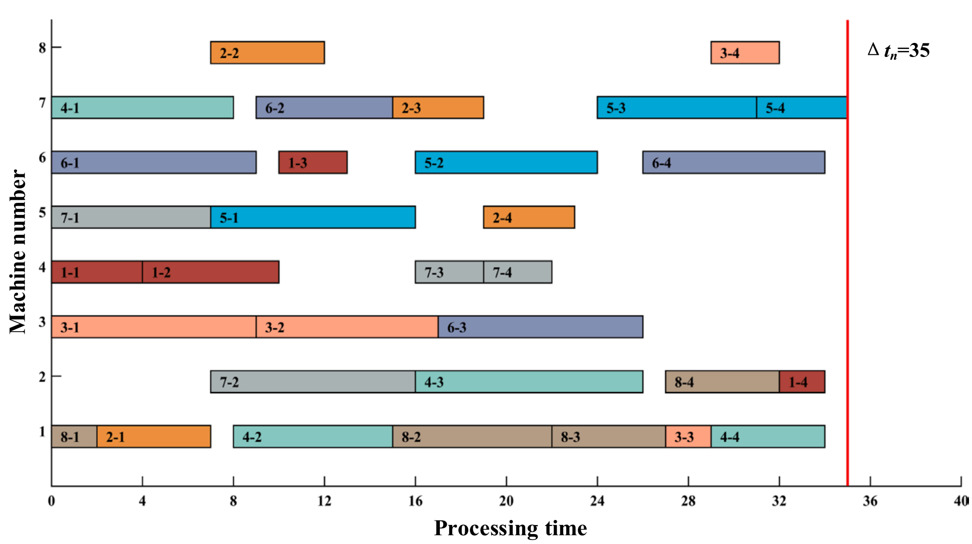

Fig. 7. Gantt chart for the original scheduled in a nondestructive mode.

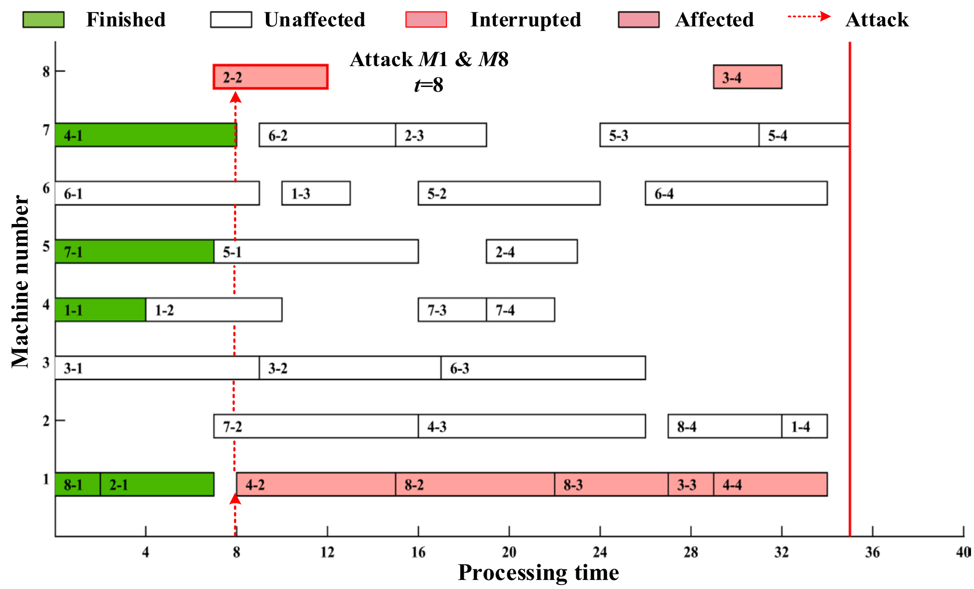

Fig. 8. Schematic diagram of the impact when an attack arrives resulting in 2 machines damage.

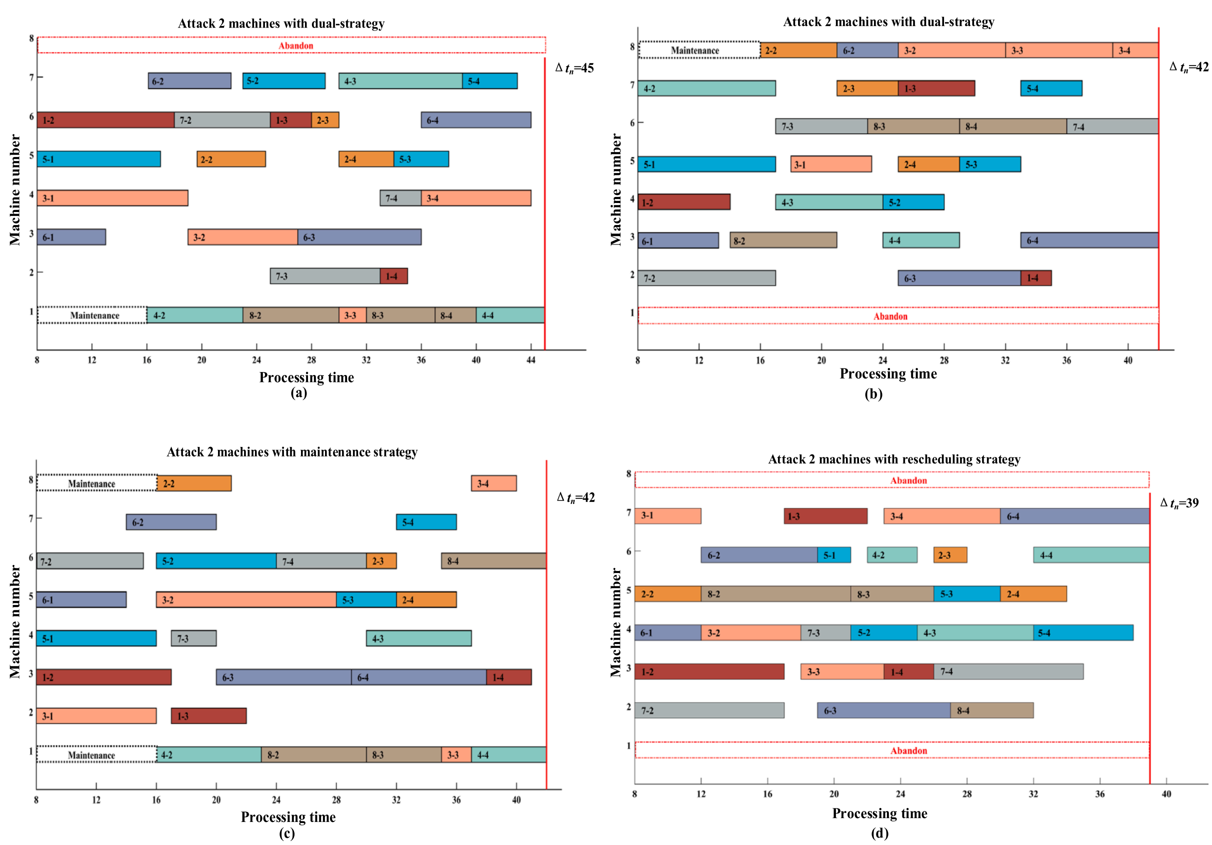

Fig. 9. Gantt chart of the recovery process under different strategies: (a) results under dual-strategy with maintenance on M_1 and rescheduling on M_8; (b) results under dual-strategy with maintenance on M_8 and rescheduling on M_1; (c) results under maintenance strategy; (d) results under rescheduling strategy.

5.2. Computational results

5.2.1. Scheduling solution found by PIOLC

In this subsection, the effectiveness of the proposed PIOLC algorithm is initially evaluated. As illustrated in Fig. 7, the optimal solution and operations sequence found by PIOLC for the 8 × 8 problem can be obtained. Specifically, the make-span value is "35″, which means that all scheduled jobs will be completed and all candidate machines are available at $t = 35$ when attacks do not launch throughout the entire schedule period.

For the above production scheduling process, assume that an attack launches at $t = 8$, causing damage to machines 1 and 8, and the corresponding operations are interrupted. Then, the current status of the system can be shown in Fig. 8, where four operations are finished (green box), one operation is interrupted (red box with red edge), six operations are affected (red box with black edge), and the remaining 20 operations

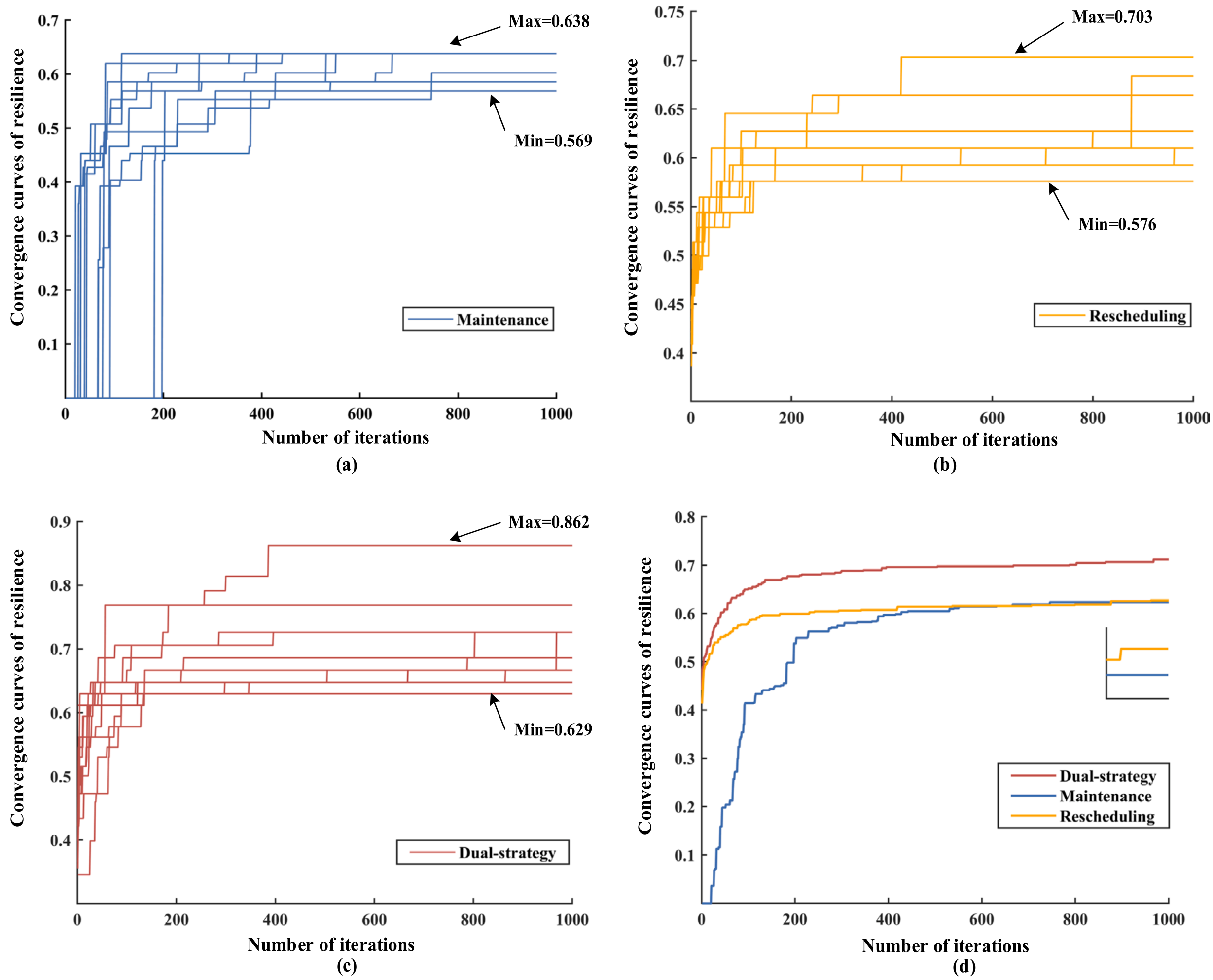

Fig. 10. The convergence curves of resilience: (a) results under the maintenance strategy; (b) results under the rescheduling strategy; (c) results under the dual-strategy; (d) comparison of mean convergence curves.

are unaffected (white box).

According to this design, recovery policies immediately executed to resume production when an attack arrives. As can be seen from Fig. 9(a)-(d), the 27 unfinished operations have been completed based on different strategies and the corresponding Gant charts of the recovery process are given. Fig. 9(a) indicates that all rest scheduled jobs are completed at $t = 45$, where maintenance strategy is applied to the machine M_1 while the 2 operations originally processed on M_8 are reassigned to other available machines. The result of the opposite strategy is given in Fig. 9(b) with a completion time $t = 42$. Since M_8 did not undertake too many tasks at first, some operations involved in rescheduling are arranged on it after the maintenance. Fig. 9(c) and (d) represent the results under maintenance and rescheduling strategies, respectively. In terms of the rescheduling strategy, the recovery time is reduced so that the last operation ends at $t = 39$. This shows the effectiveness of our proposed method in rapid recovery to a certain extent.

Table 2
Resilience convergence values under different number of machine damage.

Strategies	Descriptive statistics	Number of machine damage			
		2	3	4	5
Maintenance	Best	0.6376	0.8425	0.9179	0.6897
	Worst	0.5688	0.5315	0.4690	0.6153
	Mean	**0.6234**	**0.6585**	**0.6635**	**0.6569**
	Std Dev	0.0255	0.0966	0.1617	0.0334
Rescheduling	Best $O_{1,1}$	0.7034	0.5437	0.4860	0.4328
	$o_{1,1}$				
	Worst	0.5759	0.4713	0.3652	0.3927
	Mean	**0.6271**	**0.4980**	**0.4086**	**0.4128**
	Std Dev	0.0418	0.0210	0.0378	0.0212
Dual-strategy	Best	0.8619	0.7688	0.6664	0.4591
	Worst	0.6294	0.5759	0.4990	0.4194
	Mean	**0.7119**	**0.6547**	**0. 5997**	**0. 4333**
	Std Dev	0.0650	0.0559	0.0558	0.0183

5.2.2. *Analysis of resilience evolution*

To further demonstrate the overall superiority and effectiveness of our proposed method, the resilience evolution process of the above case is analyzed in this subsection. The convergence trends of recovery strategies on resilience optimization are shown in Fig. 10 (a)-(c). In each case, the recovery strategies are composed of maintenance, rescheduling, and a dual-strategy method. About parameter settings, we set the number of iterations to 1000, population size to 100, and each instance is performed for ten counts of experiments. Obviously, the resilience optimization can be realized based on our proposed PIOLC algorithm. It can be seen that the dual-strategy occupies a competitive achievement for reaching the maximum resilience value of 0.862. While maintenance holds the lowest terminate value among the three strategies and its optimal value is merely close to the worst value of the dual-strategy.

The comparisons of mean convergence curves are given in Fig. 10 (d) which implies overall recovery capabilities of the strategies. Clearly, the dual-strategy considering both rescheduling and maintenance obtains the best recovery results and shows a superiority of convergence rate. As

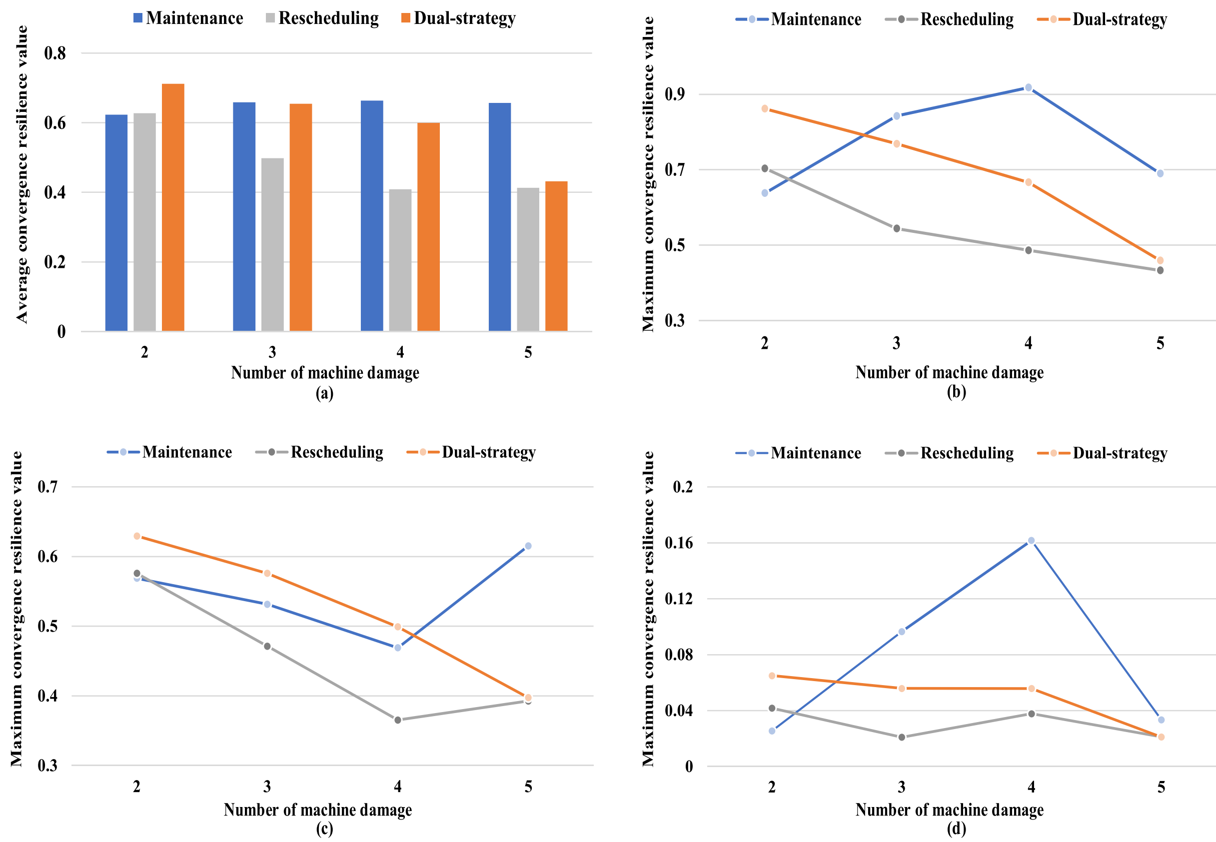

Fig. 11. Comparison of solutions under different strategies.

a result, we can conclude that the proposed framework perfectly balances the strategy's exploitation and exploration process to significantly enhance PIOLC's search capabilities and recovery effect.

5.3. Discussion

Uncertainty management is a crucial part of any systems to achieve controllable behavior. To observe the effect of external attacks on the results, this stage of the computational study investigates the distribution of solutions under the situation of different degree of damage. Another aspect involving the arrival time of an attack will be discussed in the future work.

5.3.1. Effect of damage degree

In this subsection, we investigate the performances of the above strategies on resilience optimization problems with the number of machine damages is 2, 3, 4, and 5 respectively. Based on statistical data tested for 10 times, the detailed information of the convergence value is listed in Table 2.

To intuitively observe the performances of different strategies, the solutions are compared in Fig. 11, and the followings have been found.

For the resilience objective, all of the policies make production recover effectively, proving the competitiveness of the proposed approach. Benefiting from its more diversely solution combination, the dual-strategy covers a larger proportion of ideal solutions that the other two methods when the number of damages is small. From the average and the best results compared in Fig. 11 (a) and (b), we can see that resilience values applying rescheduling strategy and dual-strategy decrease as the number of machine damages increases, while the maintenance has the opposite trend. As shown in Fig. 11 (c), the worst value of each strategy decreases with the aggravation of damages. Although the advantages of maintenance strategy are becoming more and more obvious, it is also becoming more and more unstable until the number of damages reach 5 (see Fig. 11 (d)).

5.3.2. Distribution of solutions in a dual-strategy method

To effectively support decision-making, this section further studies the distribution of solutions in a dual-strategy method. In this regard, the resilience convergence values obtained based on the combination of different strategies in the above dual-strategy method are statistically analyzed. The distributions of resilience convergence values for each condition are available in Fig. 12 by quartile graphs and the top and bottom of the blue box correspond to the upper and lower quartile lines, while the values beyond the two short black lines are extreme outliers (shown as red star signs).

Fig. 12 (a) shows that the unfinished operations processed on M_1 are involved in rescheduling, and the maintenance strategy is conducted on M_8 can make the system recover better after the two machines are damaged. From Fig. 12 (b)-(d), an obvious trend shows that with the increase of the number of machine damage, the proportion of maintenance strategy in a dual-strategy method is gradually increasing for a better resilience.

In general, the experimental results are reasonable, which indicates that our proposed method can initially provide an accurate scheme for decision-making.

6. Conclusion

In this paper, a novel resilience metric for SMS systems is proposed from the perspective of recovery oriented to time. The corresponding PIOLC optimization method is developed based on a dual-strategy recovery framework. A flexible job shop scheduling problem with 8 machines and 8 jobs is used to demonstrate the application of the metric and its corresponding dual-strategy recovery methodology. The experimental results show that the resilience metric is reasonable and that the corresponding optimization is efficient.

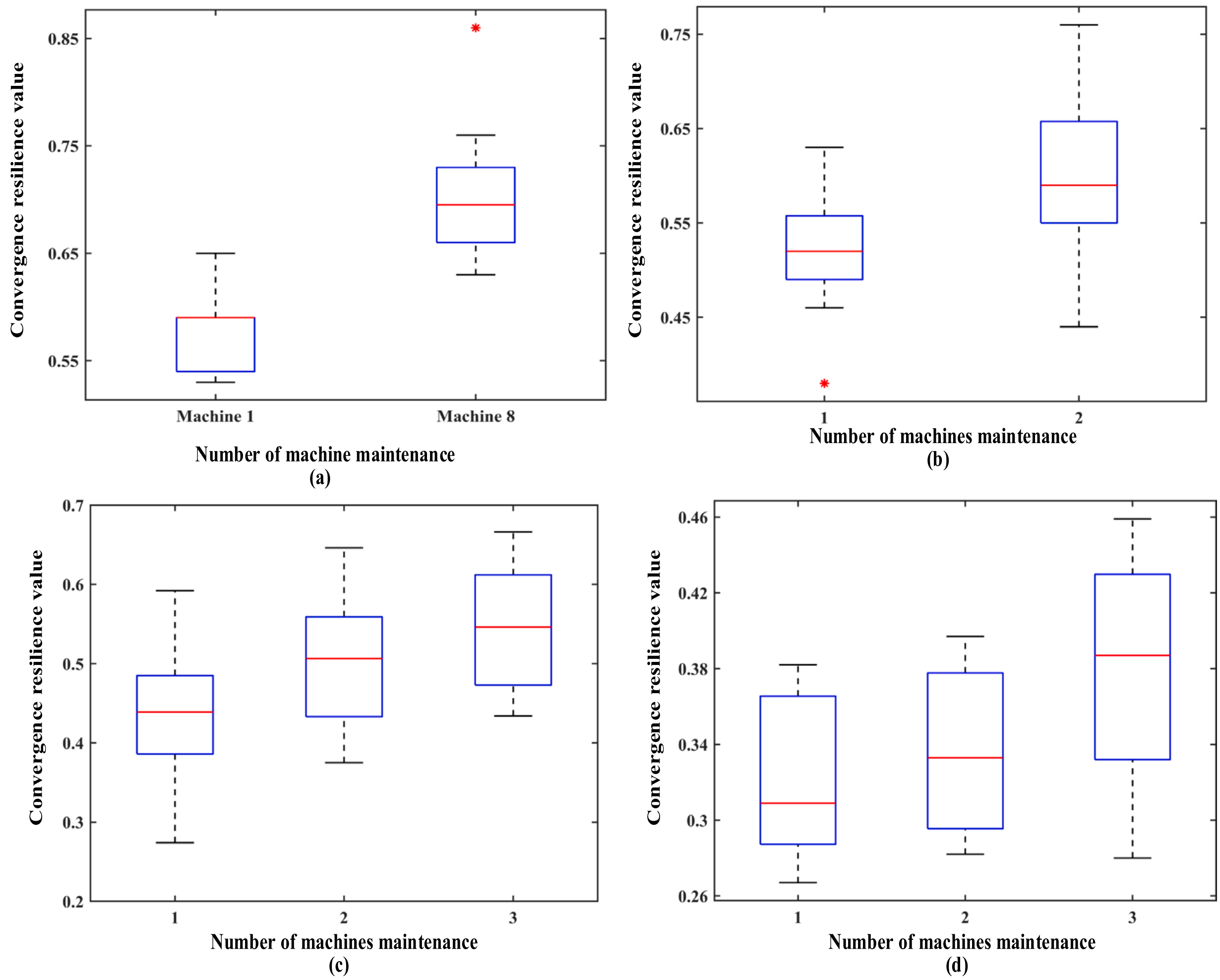

Fig. 12. Quartile graphs of the solutions under different strategy combinations: (a) 2 machines damaged; (b) 3 machines damaged; (c) 4 machines damaged; (d) 5 machines damaged.

The proposed time-based resilience metric is different from the traditional quantifiable approaches where the quantification only involves the time factor, which represents the most common but also the most critical component of SMS. This is also the main reason why it has a concise form and is easy to employ. The solution framework accompanied by a graph-based dual-strategy representation and a joint optimization model provides the possibility for the implementations and tradeoffs of rescheduling and maintenance. Owing to the enhanced PIO algorithm, where global search capabilities are strengthened by the learning strategy and parameter updating mechanism, and the diversity of the population is increased by an additional crossover operator, the optimal solution of dual-strategy recovery can be realized through resilience optimization in an acceptable way. For the case of resilience optimization, the proposed systematic framework is general and thus can be applied to the design and analysis of SMS systems for any recovery process with uncertainties causing damage to machines.

Although our proposed method showed prominent performances, simulation results based on a single case study with specific conditions can have some limitations. Due to mathematical combinations, selection of parameters, and data sources, the proposed resilience metric may have some deficiencies in scalability and flexibility. In practical application, a disruptive event can eventually cause absolute or partial damages of both production and machines. Degradation failure and human factors are also important aspects that deserve consideration.

For future research, other rescheduling and maintenance policies can be investigated to deal with real-time attacks. An integrated paradigm considering sustainability, resilience, and lean practices accompanied with multiobjective optimization approach can be further studied. Comparisons among PIOLC, data-driven methods, and AI-based techniques are necessary. Uncertainties can be extended to arrival time and different failure modes. Human factors need to be precisely modeled in the proposed decision-making scheme. Another promising direction would be the investigation of similarities and differences between our proposed work and the concept of "Maintenance-free Factory" as well as reconfigurable manufacturing systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is collaboratively funded by the National Key Research and Development Program of China [No. 2019YFE0105100] and Research Council of Norway [AutoPRO, No. 309628].

References

[1] Wang B.C., Tao F., Fang X.D., Liu C., Liu Y.F., Freiheit T. Smart manufacturing and intelligent manufacturing: A comparative review. Engineering 2020.
[2] Chhetri SR, Rashid N, Faezi S, Faruque MAA. Security trends and advances in manufacturing systems in the era of industry 4.0. Proc 36th Int Conf Comput-Aided Des 2017:1039–46.

[3] Serrano-Ruiz JC, Mula J, Poler R. Smart manufacturing scheduling: a literature review. J Manuf Syst 2021;56:265–87.
[4] Tuptuk N, Hailes S. Security of smart manufacturing systems. J Manuf Syst 2018;47:93–106.
[5] Lee PC, Chen SH, Lin YS, Su HN. Toward a better understanding on technological resilience for sustaining industrial development. IEEE Trans Eng Manag 2019;66(3):398–411.
[6] Hosseini S, Khaled AA, Sarder MD. A general framework for assessing system resilience using Bayesian networks: a case study of sulfuric acid manufacturer. J Manuf Syst 2016;41:211–27.
[7] Su QL, Moreno M, Ganesh S, Reklaitis GV, Nagy ZK. Resilience and risk analysis of fault-tolerant process control design in continuous pharmaceutical manufacturing. J Loss Prev Process Ind 2018;55:411–22.
[8] Zhang D, Xie M, Yan H, Liu Q. Resilience dynamics modeling and control for a reconfigurable electronic assembly line under spatio-temporal disruptions. J Manuf Syst 2021.
[9] Naghshineh B, Carvalho H. The implications of additive manufacturing technology adoption for supply chain resilience: a systematic research and review. Int J Prod Econ 2022;247:108387.
[10] Romero D, Stahre J. Towards the resilient operator 5.0: the future of work in smart resilient manufacturing system. Procedia CIRP 2021;104:1089–94.
[11] Kusiak A. Open manufacturing: a design-for-resilience approach. Prod Res 2020;58(15):4647–58.
[12] Conz E, Magnani G. A dynamic perspective on the resilience of firms: a systematic literature review and a framework for future research. Eur Manag J 2020;38(3):400–12.
[13] Goldbeck N, Angeloudis P, Ochieng W. Optimal supply chain resilience with consideration of failure propagation and repair logistics. Transp Res Part E 2020;133:101830.
[14] Carvalho H, Naghshineh B, Govindan K, Cruz-Machado V. The resilience of on-time delivery to capacity and material shortages: an empirical investigation in the automotive supply chain. Comput Ind Eng 2022;171:108375.
[15] Henry D, Ramirez-Marquez JE. Generic metrics and quantitative approaches for system resilience as function of time. Reliab Eng Syst Saf 2012;99:114–22.
[16] Romero D, Stahre J, larsson L, Rönnbäck AÖ. Building manufacturing resilience through production innovation. IEEE Int Conf Eng Technol Innov 2021.
[17] Vieira GE, Herrmann JW, Lin AE. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. J Sched 2003;6:39–62.
[18] Gao GB, Zhou DM, Tang H, Hu X. An intelligent health diagnosis and maintenance decision-making approach in smart manufacturing. Reliab Eng Syst Saf 2021;2021(216):107965.
[19] Salido MA, Escamilla J, Barber F, Giret A. Rescheduling in job-shop problems for sustainable manufacturing systems. J Clean Prod 2017;162:S121–32.
[20] Gao KZ, Yang FJ, Zhou MC, Pan QK, Suganthan PN. Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm. IEEE Trans Cybern 2019;49(5):1944–55.
[21] Xia TB, Shi G, Si GJ, Du SC, Xi LF. Energy-oriented joint optimization of machine maintenance and tool replacement in sustainable manufacturing. J Manuf Syst 2021;59:261–71.
[22] Tian Y, Zhu C, Chang Q, Wang JF. Imperfect corrective maintenance scheduling for energy efficient manufacturing systems through online task allocation method. J Manuf Syst 2019;53:282–90.
[23] Li KX, Deng QW, Zhang LK, Fan Q, Gong GL, Ding S. An effective MCTS-based algorithm for minimizing makespan in dynamic flexible job shop scheduling problem. Comp Ind Eng 2021;155:107211.
[24] Hu JW, Jiang ZH, Liao HT. Joint optimization of job scheduling and maintenance planning for a two-machine flow shop considering job-dependent operating condition. J Manuf Syst 2020;57:231–41.
[25] Yang HB, Li WC, Wang B. Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. Reliab Eng Syst Saf 2021;214:107713.
[26] Gahleb M, Taghipour S, Zolfagharinia H. Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance. J Manuf Syst 2021;61:424–49.
[27] Chen XH, An YJ, Zhang ZY, Li YH. An approximate nondominated sorting genetic algorithm to integrate optimization of production scheduling and accurate maintenance based on reliability intervals. J Manuf Syst 2020;54:227–41.
[28] Chari A. , Duberg J.V. , Lindahl E. , Stahre J., Despeisse M. , Sundin E. , , et al. Swedish Manufacturing Practices Towards a Sustainability Transition in Industry 4.0: A Resilience Perspective. Proceedings of the ASME 2021 16th International Manufacturing Science and Engineering Conference 2021; Virtual, Online.
[29] Duan HB, Qiao PX. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int J Intell Comput Cybern 2014;7(1):24–37.
[30] Duan HB, Huo MZ, Shi YH. Limit-cycle-based mutant multiobjective pigeon-inspired optimization. IEEE Trans Evolut Comput 2020;24(5):948–59.
[31] Hai XS, Wang ZL, Feng Q, Ren Y, Sun B, Yang DZ. A novel adaptive pigeon-inspired optimization algorithm based on evolutionary game theory. Sci China Inf Sci 2021;64(3):1–2.
[32] Wu XL, Shen XL, Li CB. The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. Comput Ind Eng 2019;135:1004–24.
[33] Fu XY, Chan FTS, Niu B, Chung NSH, Qu T. A multi-objective pigeon inspired optimization algorithm for fuzzy production scheduling problem considering mould maintenance. Sci China Inf Sci 2019;62(7):010202.
[34] Duan HB, Qiu HX. Advancements in pigeon-inspired optimization and its variants. Sci China Inf Sci 2019;62:070201.
[35] Serrano-Ruiz JC, Mula J, Poler R. Development of a multidimensional conceptual model for job shop smart manufacturing scheduling from the Industry 4.0 perspective. J Manuf Syst 2022;63:185–202.
[36] Moencks M, Roth E, Bohne T, Romero D, Stahre J. Augmented Workforce Canvas: a management tool for guiding human-centric, value-driven human-technology integration in industry. Comput Ind Eng 2022;163:107803.
[37] Feng Q, Hai XS, Huang BQ, Zuo Z, Ren Y, Sun B, et al. An agent-based reliability and performance modeling approach for multistate complex human-machine systems with dynamic behavior. IEEE Access 2019;7:135300–11.
[38] Fan DM, Ren Y, Feng Q, Liu YL, Wang ZL, Lin J. Restoration of smart grids: Current status, challenges, and opportunities. Renew Sustain Energy Rev 2021;143:110909.
[39] Cai BP, Xie M, Liu YH, Liu YL, Feng Q. Availability-based engineering resilience metric and its corresponding evaluation methodology. Reliab Eng Syst Saf 2018;172:216–24.
[40] Feng Q, Zhao XJ, Fan DM, Cai BP, Liu YQ, Ren Y. Resilience design method based on meta-structure: a case study of offshore wind farm. Reliab Eng Syst Saf 2019;186:232–44.
[41] 2019 World Manufacturing Forum Report, Skills for the Future of Manufacturing.