frontiers | Frontiers in Environmental Science

TYPE Original Research
PUBLISHED 04 November 2022
DOI 10.3389/fenvs.2022.1059451

OPEN ACCESS

EDITED BY
Uzair Aslam Bhatti,
Hainan University, China

REVIEWED BY
Yongchao Luo,
South China University of Technology, China
Hongyan Shi,
Shenzhen University, China
Tao Ku,
Shenyang Institute of Automation (SIA) (CAS), China

*CORRESPONDENCE
Hao Tang,
melineth@hainanu.edu.cn

SPECIALTY SECTION
This article was submitted to Environmental Informatics and Remote Sensing,
a section of the journal
Frontiers in Environmental Science

RECEIVED 01 October 2022
ACCEPTED 19 October 2022
PUBLISHED 04 November 2022

CITATION
Wang S, Li J, Tang H and Wang J (2022), CEA-FJSP: Carbon emission-aware flexible job-shop scheduling based on deep reinforcement learning.
Front. Environ. Sci. 10:1059451.
doi: 10.3389/fenvs.2022.1059451

CEA-FJSP: Carbon emission-aware flexible job-shop scheduling based on deep reinforcement learning

Shiyong Wang[1], Jiaxian Li[1], Hao Tang[2]* and Juan Wang[3]

[1]School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China, [2]School of Information and Communication Engineering, Hainan University, Haikou, Hainan, China, [3]School of Electronics and Communication, Guangdong Mechanical & Electronical Polytechnic, Guangzhou, China

Currently, excessive carbon emission is causing visible damage to the ecosystem and will lead to long-term environmental degradation in the future. The manufacturing industry is one of the main contributors to the carbon emission problem. Therefore, the reduction of carbon emissions should be considered at all levels of production activities. In this paper, the carbon emission as a parvenu indicator is considered parallelly with the nobleman indicator, makespan, in the flexible job-shop scheduling problem. Firstly, the carbon emission is modeled based on the energy consumption of machine operation and the coolant treatment during the production process. Then, a deep reinforcement learning-based scheduling model is proposed to handle the carbon emission-aware flexible job-shop scheduling problem. The proposed model treats scheduling as a Markov decision process, where the scheduling agent and the scheduling environment interact repeatedly via states, actions, and rewards. Next, a deep neural network is employed to parameterize the scheduling policy. Then, the proximal policy optimization algorithm is conducted to drive the deep neural network to learn the objective-oriented optimal mapping from the states to the actions. The experimental results verify that the proposed deep reinforcement learning-based scheduling model has prominent optimization and generalization abilities. Moreover, the proposed model presents a nonlinear optimization effect over the weight combinations.

KEYWORDS

smart manufacturing, production scheduling, deep reinforcement learning, carbon emission, multi-objective optimization

1 Introduction

Production scheduling is a subclass of combinational optimization problems aiming to sequence jobs to machines toward the optimization of one or more scheduling objectives (Fernandes et al., 2022). Production scheduling can be classified into many types according to its inherent properties. For example, the job-shop scheduling problem (JSSP) specifies that one operation can only be processed by one machine (Zhang et al.,

2019), while the flexible job-shop scheduling problem (FJSP) allows multiple candidate machines to process an operation (Brucker and Schlie, 1990). The frequently adopted scheduling objectives related to economic benefits include makespan, tardiness, and machine utilization (Allahverdi et al., 2008). In recent years, the steady deterioration of environmental problems (Bhatti et al., 2021; Bhatti et al., 2022a; Bhatti et al., 2022b), such as pollution and climate change, has raised the awareness of environmental protection. Hence, environmental indicators, especially energy consumption and carbon emission, are a growing concern in production scheduling (Gao et al., 2020). Therefore, the FJSP is formulated as a multi-objective optimization problem considering both economic benefit and environmental effect.

The heuristic and meta-heuristic algorithms have been widely applied to achieve multi-objective scheduling. In terms of heuristic algorithms, Zhang et al. (2022) proposed a greedy algorithm and an elite strategy to solve FJSP with the objectives of minimizing both makespan and total energy consumption. Xu et al. (2021) proposed three delayed routing strategies to optimize energy efficiency and mean tardiness in dynamic FJSP. In terms of meta-heuristic algorithms, the multi-objective genetic algorithm (GA) is the most popular scheme due to its excellent global optimization ability and convergence performance (Li and Wang, 2022). Several GA-based algorithms have been proposed to improve search efficiency for minimizing makespan and total energy consumption (Mokhtari and Hasani, 2017; Dai et al., 2019) and to determine machine start/stop time and speed level to save energy (Wu and Sun, 2018). Moreover, none-GA based algorithms including the frog-leaping algorithm (Lei et al., 2017) and the grey wolf algorithm (Luo et al., 2019) are also available for multi-objective scheduling.

However, the above-mentioned scheduling algorithms lack generalization ability (Han and Yang, 2021). To solve an FJSP instance that is different from the solved ones in terms of parameters such as the number of jobs and machines, the existing heuristic algorithms generally require the development of new scheduling rules while the meta-heuristic algorithms require considerable iterative computation time to obtain high-quality scheduling solutions. In contrast, deep reinforcement learning (DRL) based (Arulkumaran et al., 2017) production scheduling can learn and generalize the knowledge from the training samples to new problems. Therefore, the trained DRL models can be applied to different scheduling scenarios to produce satisfactory scheduling solutions in a reasonable computation time. Qu et al. (2016) and van Ekeris et al. (2021) stated that DRL could discover basic heuristic behaviors for production scheduling from scratch, providing a kind of optimization-capable, scalable, and real-time scheduling methods.

Numerous studies have utilized the generalization ability of DRL to solve different-scale production scheduling problems (Ren et al., 2020; Zhang et al., 2020; Monaci et al., 2021; Ni et al., 2021; Park et al., 2021). However, these studies focused on either the single objective JSSP (Han and Yang, 2020; Liu et al., 2020; Zhao et al., 2021; Zeng et al., 2022) or the flow-shop scheduling problem (FSSP) (Pan et al., 2021; Yan et al., 2022). The multi-objective FJSPs have been seldom addressed (Lang et al., 2020; Luo et al., 2021). Furthermore, among the few studies addressing the DRL-based multi-objective FJSP, even fewer studies cared about environmental objectives (Naimi et al., 2021; Du et al., 2022). Therefore, the development of DRL-based methods for solving FJSP is still in the initial stage and not yet systematic (Luo, 2020; Feng et al., 2021; Liu et al., 2022).

In summary, the existing DRL-based methods for FJSP receive less attention compared with those for JSSP. Moreover, most of the studies preferred the optimization of single or multiple economic objectives to the optimization of environmental objectives. Although some studies have attempted to minimize total energy consumption or electricity cost, minimizing carbon emissions has not been yet explicitly considered. Furthermore, a few studies integrated a DRL model with a meta-heuristic algorithm to solve the multi-objective FJSP. However, the DRL model was used as an auxiliary tool to assist the meta-heuristic algorithm to improve search efficiency. To resolve the above-mentioned technical limitation, this paper proposes a DRL-based scheduling method to handle FJSP to minimize both makespan and total carbon emission. The main contributions of this study are listed as follows.

1) The classical FJSP is extended to a carbon emission-aware flexible job-shop scheduling problem (CEA-FJSP), where a carbon emission accounting model is formulated based on the energy consumption of machine operation and coolant treatment during the production process.
2) An intelligent DRL-based scheduling model is developed to directly generate feasible scheduling solutions for CEA-FJSP without extra searching. The solving process is modeled as a Markov decision process (MDP) including generic productive state features, a scheduling rule-based action space, and a composite reward function.
3) The scheduling policy is parameterized by a deep neural network (DNN), that is, optimized by the proximal policy optimization (PPO) algorithm to establish the mapping from the states to the actions.
4) The experimental results on various benchmarks demonstrate that the proposed DRL scheduling model has prominent optimization and generalization abilities. Moreover, the proposed model presents a nonlinear optimization effect over the weight combinations.

The remainder of this paper is organized as follows. The mathematical model of the CEA-FJSP is formulated in Section 2. The DRL scheduling model is described in Section 3. Section 4

TABLE 1 Notations for CEA-FJSP.

Notation	Description
i, i'	Indices of jobs, $i, i' = 1, 2, \ldots, n$
j, j'	Indices of operations, $j, j' = 1, 2, \ldots, n_i$
k	Index of machines, $k = 1, 2, \ldots, m$
n	Total number of jobs
m	Total number of machines
$\mathcal{I}$	Job set
$\mathcal{M}$	Machine set
$\mathcal{M}_{ij}$	Candidate machines for operation O_{ij}
J_i	The ith job
O_{ij}	The jth operation of job J_i
M_k	The kth machine
n_i	The number of operations of job J_i
t_{ijk}	The processing time of operation O_{ij} required by machine M_k
S_{ij}	The start time of operation O_{ij}
C_{ij}	The completion time of operation O_{ij}
t_k^{idle}	The idle time of machine M_k
p_{ijk}	The processing power of operation O_{ij} required by machine M_k
p_k^{idle}	Idle power of machine M_k
T_k	The replacement cycle of coolant required by machine M_k
L_k	The maximum usage of coolant required by machine M_k in a cycle
α_e	The carbon emission factor of electrical energy consumption
α_f	The carbon emission factor of coolant energy consumption
CE_p	Carbon emission from electrical energy consumption of machine in processing state
CE_r	Carbon emission from electrical energy consumption of machine in idle state
CE_f	Total carbon emission from energy consumption of coolant treatment
TCE	Total carbon emission
$C_{\max}$	Makespan
x_{ijk}	A binary variable, that is, equal to 1 if O_{ij} is assigned to machine M_k and 0 otherwise
$y_{iji'j',k}$	A binary variable, that is, equal to 1 if O_{ij} is a predecessor of $O_{i'j'}$ on machine M_k and 0 otherwise

presents the experimental results and Section 5 concludes the study.

2 Problem formulation

This section mathematically describes the conditions and constraints of the CEA-FJSP. There are n jobs belonging to a job set $\mathcal{I} = \{J_1, J_2, \ldots, J_n\}$ to be processed by m machines belonging to a machine set $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. A job J_i consists of n_i operations, where O_{ij} denotes the j th operation of J_i. The operations of the same job J_i must be processed in a specific order, i.e., $O_{i1} \rightarrow O_{i2} \cdots \rightarrow O_{in_i}$. The operation O_{ij} can be processed by one or more machines forming an operation-specific candidate machine set $\mathcal{M}_{ij} \subseteq \mathcal{M}$. The time and the power that the machine $M_k \in \mathcal{M}_{ij}$ requires to process the operation O_{ij} are denoted as t_{ijk} and p_{ijk}, respectively. The machine M_k also requires coolant during processing and constant lower power consumption in an idle state. The scheduling for CEA-FJSP aims to obtain an optimal scheduling solution to minimize both makespan and carbon emission, by determining a machine M_k from $\mathcal{M}_{ij}$, the start time S_{ij}, and the completion time $C_{ij} = S_{ij} + t_{ijk}$ for each operation O_{ij}. Furthermore, the following constraints and assumptions should be satisfied:

1) The operations of the same job should be processed following the defined operation precedence.
2) A machine can only process one operation at a time.
3) An operation should be processed without interruption.
4) A machine processes an operation with constant processing power.
5) All machines turn on at the start of the scheduling.
6) The transportation time of jobs and the setup time of machines are negligible.

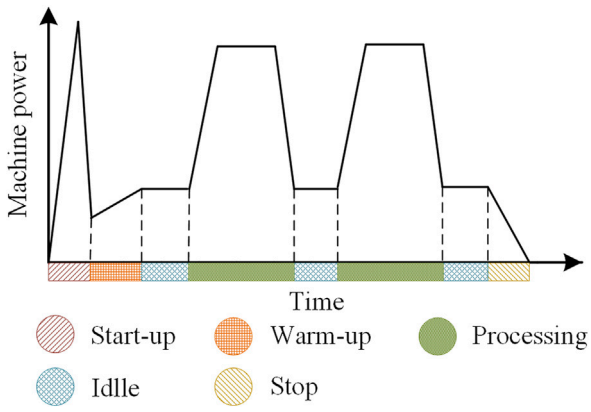

FIGURE 1
Power variation of five machine working modes.

Based on the above description, a carbon emission accounting model is formulated firstly to identify the main sources and specific computation of carbon emission in CEA-FJSP. Then, the mathematical model of the CEA-FJSP is established. Table 1 lists the notations used in the models.

2.1 Carbon emission accounting model

Carbon emission is produced directly or indirectly by various manufacturing links, such as raw materials consumption, machine operation, transportation, and metal debris treatment (Gutowski et al., 2005). In this paper, the electrical energy consumption of machine operation and the energy consumption of coolant treatment are identified as the main carbon emission sources in CEA-FJSP.

2.1.1 Carbon emission from machine operation

Generally, a machine experiences five working modes in a duty cycle: start-up, warm-up, processing, idle, and stop. Each mode requires a different power level as shown in Figure 1. The modes of start-up, warm-up, and stop appear only once in a duty cycle and the energy consumption in these modes is only related to machine properties rather than scheduling. In contrast, the processing and idle modes tend to alternately appear multiple times. Therefore, only the carbon emission in processing and idle modes are considered in scheduling.

Under processing mode, the carbon emission CE_p is calculated as:

$$CE_p = \alpha_e W_p \tag{1}$$

where W_p is the total electrical energy consumption of all machines under processing mode and is expressed as:

$$W_p = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i} x_{ijk} p_{ijk} t_{ijk} \tag{2}$$

Under idle mode, the carbon emission CE_r is calculated as:

$$CE_r = \alpha_e W_r \tag{3}$$

where W_r is the total electrical energy consumption of all machines under idle mode and is expressed as:

$$W_r = \sum_{k=1}^{m} p_k^{idle} t_k^{idle} \tag{4}$$

2.1.2 Carbon emission from coolant treatment

The coolant is used to reduce the cutting temperature and tool wear and prevent the workpiece from being deformed by heat. The coolant needs to be replaced periodically and the treatment process consumes energy, indirectly producing carbon emissions. To simplify the calculation, it is assumed that the coolant flow rate remains unchanged for the same machine regardless of the processed operations. Hence, the carbon emission of coolant treatment CE_f can be calculated as:

$$CE_f = \alpha_f \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i} x_{ijk} \frac{t_{ijk}}{T_k} L_k \tag{5}$$

The total carbon emission TCE during scheduling adds up as:

$$TCE = CE_p + CE_r + CE_f = \alpha_e \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i} x_{ijk} p_{ijk} t_{ijk} + \alpha_e \sum_{k=1}^{m} p_k^{idle} t_k^{idle} + \alpha_f \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i} x_{ijk} \frac{t_{ijk}}{T_k} L_k \tag{6}$$

2.2 CEA-FJSP formulation

The CEA-FJSP is a multi-objective optimization problem, considering both economic and environmental benefits. The scheduling objectives are to simultaneously minimize $C_{\max} = \max\{C_{in_i} | i = 1, 2, \ldots, n\}$ and TCE. Therefore, the mathematical model of CEA-FJSP is formulated as:

$$\min f = \min\{w_1 C_{\max} + w_2 TCE\} \tag{7}$$

$$s.t. \begin{cases} C_{\max} \geq C_{ij}, \forall i, j & \text{(a)} \\ C_{ij} = S_{ij} + t_{ijk}, S_{ij} \geq 0, \forall i, j, k & \text{(b)} \\ \sum_{M_k \in \mathcal{M}_{ij}} x_{ijk} = 1, \forall i, j & \text{(c)} \\ S_{i,j+1} \geq C_{ij}, \forall i, j & \text{(d)} \\ C_{i'j'} - C_{ij} \geq t_{i'j'k},\ y_{iji'j'k} = 1 & \text{(e)} \end{cases} \tag{8}$$

Eq. 7 shows that the objective function minimizes the weighted sum of $C_{\max}$ and TCE, converting the multi-objective optimization problem into a single-objective optimization problem, where w_1 and w_2 are the weights corresponding to $C_{\max}$ and TCE, respectively. Eq. 8 shows the five constraints.

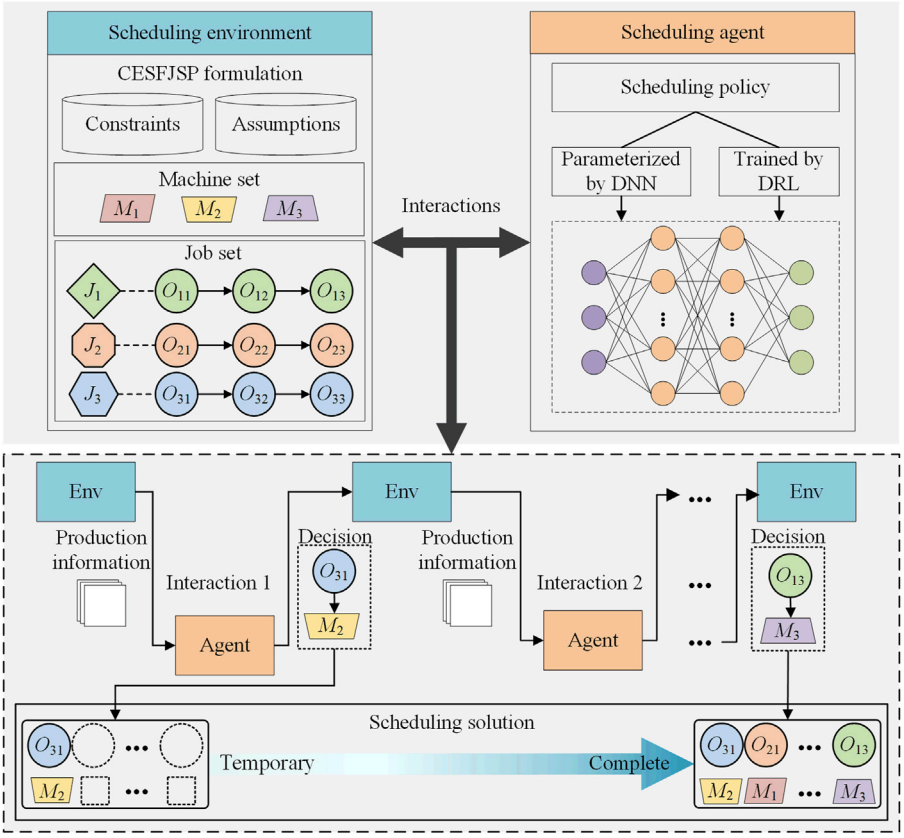

FIGURE 2
Framework of the DRL scheduling model for CEA-FJSP.

Constraint (a) in Eq. 8 describes the relationship between makespan and the operation completion time. Constraint (b) ensures that the operation completion time is equal to the sum of the start time and the processing time. Constraint (c) specifies that an operation can be assigned to and processed by only one machine. Constraint (d) guarantees the precedence constraint between the operations of the same job. Constraint (e) shows that a machine can process only one operation at a time.

3 Deep reinforcement learning scheduling modeling

This section proposes a DRL scheduling model for handling CEA-FJSP. Figure 2 shows the framework of the proposed DRL scheduling model. The scheduling environment is an instance of CEA-FJSP initialized with the assumptions and constraints described in Section 2. The scheduling agent embeds a scheduling policy parameterized by a DNN and trained by a DRL algorithm. The agent interacts repeatedly with the environment. In each interaction, the scheduling agent selects an operation and assigns it to a machine, based on the information extracted from the scheduling environment.

The determined operations are queued in a temporary scheduling solution, which is a sequence intuitively describing the precedence of operations. The temporary scheduling solution is turned into a complete and feasible scheduling solution when all operations are determined. Therefore, the scheduling process of a CEA-FJSP instance features an MDP consisting of state, action, and reward. Lastly, the MDP is optimized using a DRL algorithm resulting in a DRL scheduling model.

3.1 Markov decision process formulation

An MDP mainly includes three components: state, action, and reward. A complete decision-making process of MDP is called an episode, consisting of $T = \sum_{i=1}^{n} n_i$ decision steps in CEA-FJSP, where one decision step corresponds to one interaction. At the decision step t, the scheduling agent perceives the state s_t of the scheduling environment. Then, the state features are fed into the scheduling policy that in turn selects an action a_t. After the

TABLE 2 Statistic-based state features.

Production statistic information	Index	State feature
Completion rate of a job (CRJ): $CRJ = \{\frac{SO_i(t)}{n_i} \mid J_i \in \mathcal{I}\}$	f_{t1}	$\text{mean}(CRJ)$
	f_{t2}	$\text{std}(CRJ)$
Machine utilization of a machine (MU): $MU = \{\frac{\sum_{i=1}^{n}\sum_{j=1}^{SO_i(t)} x_{ijk}t_{ijk}}{\max(\{C_{i,SO_i(t)} \mid i=1,2,\ldots,n\})} \mid M_k \in \mathcal{M}\}$	f_{t3}	$\text{mean}(MU)$
	f_{t4}	$\text{std}(MU)$
Current carbon emission produced by a machine (MCE): $MCE = \{\alpha_e[\sum_{i=1}^{n}\sum_{j=1}^{SO_i(t)}(x_{ijk}p_{ijk}t_{ijk} + p_k^{idle}t_k^{idle})] + \alpha_f\sum_{i=1}^{n}\sum_{j=1}^{SO_i(t)} x_{ijk}\frac{t_{ijk}}{T_k}L_k \mid M_k \in \mathcal{M}\}$; Number of operations processed on a machine (MN): $MN = \{\sum_i^n\sum_{j=1}^{SO_i(t)} x_{ijk} \mid M_k \in \mathcal{M}\}$	f_{t5}	$\frac{\text{mean}(MCE)}{\max(MCE)}$
	f_{t6}	$\frac{\text{mean}(MN/MCE)}{\max(MN/MCE)}$
Average processing time of the current eligible operation of an uncompleted job (CPT): $CPT = \{\bar{t}_{i,SO_i(t)+1} \mid O_{i,SO_i(t)+1} \in \boldsymbol{O}^{\boldsymbol{US}}(t),\ i = 1, 2, \ldots, n\}$	f_{t7}	$\frac{\min(CPT)}{\text{mean}(CPT)}$
Average processing power of the current eligible operation of an uncompleted job (CPP): $CPP = \{\bar{p}_{i,SO_i(t)+1} \mid O_{i,SO_i(t)+1} \in \boldsymbol{O}^{\boldsymbol{US}}(t),\ i = 1, 2, \ldots, n\}$	f_{t8}	$\frac{\min(CPP)}{\text{mean}(CPP)}$
Average remaining processing time of an uncompleted job (RPT): $RPT = \{\sum_{j=SO_i(t)+1}^{n_i}\bar{t}_{ij} \mid O_{i,SO_i(t)+1} \in \boldsymbol{O}^{\boldsymbol{US}}(t),\ i = 1, 2, \ldots, n\}$	f_{t9}	$\frac{\text{mean}(RPT)}{\max(RPT)}$
Average remaining processing power of an uncompleted job (RPP): $RPP = \{\sum_{j=SO_i(t)+1}^{n_i}\bar{p}_{ij} \mid O_{i,SO_i(t)+1} \in \boldsymbol{O}^{\boldsymbol{US}}(t),\ i = 1, 2, \ldots, n\}$	f_{t10}	$\frac{\text{mean}(RPP)}{\max(RPP)}$

execution of the action a_t, an unscheduled operation is selected and assigned to a candidate machine. Hence, the selected operation becomes scheduled. After that, the scheduling environment releases a reward r_t to reflect the change of the scheduling objectives, as well as updates to a new state s_{t+1} ready for the next interaction.

3.1.1 State representation

The state is the basis of decision making and should provide adequate information about the scheduling environment. The number of scheduled operations of job J_i at the decision step t is denoted as $SO_i(t)$. The operations of all the jobs in a scheduling instance are divided into two subsets: $\boldsymbol{O}^{\boldsymbol{S}}(t) = \{O_{ij} \mid 1 \le i \le n, 1 \le j \le SO_i(t)\}$ and $\boldsymbol{O}^{\boldsymbol{US}}(t) = \{O_{ij} \mid 1 \le i \le n, SO_i(t) < j \le n_i\}$. Therefore, the completion time C_{ij} can be determined for the operations in the subset $\boldsymbol{O}^{\boldsymbol{S}}(t)$ while the average processing time $\bar{t}_{ij} = \underset{M_k \in \mathcal{M}_{ij}}{\text{mean}}(t_{ijk})$ and the average processing power $\bar{p}_{ij} = \underset{M_k \in \mathcal{M}_{ij}}{\text{mean}}(p_{ijk})$ can be calculated for the operations in the subset $\boldsymbol{O}^{\boldsymbol{US}}(t)$.

A statistic-based representation is adopted to define state features using the dynamic attributes of jobs and machines. Table 2 lists the proposed statistic-based state features. It can be seen from the table that the state is a vector consisting of ten features $\{f_{t1}, f_{t2}, \ldots, f_{t10}\}$ maintaining a fixed size, which can avoid dimension disaster in large-scale problems. Moreover, the values of the state features are in the range of [0, 1], which can speed up the training process and can be generalized to problems with different configurations.

3.1.2 Action space

Actions are used to update the scheduling environment, playing a significant role in the quality of scheduling solutions. In the CEA-FJSP, one decision contains two parts: operation selection and machine assignment. Due to the precedence constraint, a job has at most one feasible operation that can be selected at a decision step. Hence, the operation selection can be simplified as the job selection. In this paper, six job selection rules and four machine assignment rules are adopted as shown in Table 3. Nine scheduling rules, $\{SR_i \mid i = 1, 2, \ldots, 9\}$, are then constructed as follows: SR_1 = {JSPT, MMAXP}, SR_2 = {JSPT, MMINU}, SR_3 = {JLPT, MMAXP}, SR_4 = {JLPT, MMINU}, SR_5 = {JMOR, MMINP}, SR_6 = {JECT, MMAXP}, SR_7 = {JMINP, MMINU}, SR_8 = {JMINP, MSPT}, SR_9 = {JMAXP, MMINU}. It indicates that a scheduling rule is a couple of a job selection rule and a machine assignment rule. The scheduling rules are called actions in MDP. Thus, the action space consists of nine elements.

3.1.3 Reward function

As shown in Eq. 7, minimizing $C_{\max}$ and TCE are the two scheduling objectives considered in the CEA-FJSP. However, since the scheduling solution is incomplete during the scheduling, the two performance indicators cannot be resolved until the end of scheduling. In other words, the actual values of $C_{\max}$ and TCE can be calculated only once per episode. Consequently, if the actual makespan and carbon emission values are used as rewards, the immediate reward will be quite sparse and cause difficulties in the convergence of the DRL algorithm.

However, the completion time and carbon emission of the scheduled operations can be used as rewards and determined as:

$$r_t^{CT} = CT(t) - CT(t+1) \tag{9}$$

$$r_t^{CE} = CE(t) - CE(t+1) \tag{10}$$

TABLE 3 Job selection and machine assignment rules.

Classification	Acronym	Description
Job selection rule	JSPT	Selecting a job with the shortest processing time
	JLPT	Selecting a job with the longest processing time
	JMOR	Selecting a job with the most remaining operations
	JECT	Selecting a job with the earliest completion time
	JMINP	Selecting a job with the minimum processing power
	JMAXP	Selecting a job with the maximum processing power
Machine assignment rule	MSPT	Assigning to a machine with the shortest processing time
	MMINP	Assigning to a machine with the minimum processing power
	MMAXP	Assigning to a machine with the maximum processing power
	MMINU	Assigning to a machine with the minimum utilization rate

where: $CT(t)$ is the current maximum completion time of jobs at decision step t:

$$CT(t) = \max\left(\left\{C_{i,SO_i(t)} \middle| i = 1, 2, \ldots, n\right\}\right) \tag{11}$$

$CE(t)$ is the currently produced carbon emission at decision step t:

$$CE(t) = \alpha_e \left[\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{SO_i(t)}\left(x_{ijk}p_{ijk}t_{ijk} + p_k^{idle}t_k^{idle}\right)\right] + \alpha_f \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{SO_i(t)} x_{ijk}\frac{t_{ijk}}{T_k}L_k \tag{12}$$

r_t^{CT} and r_t^{CE} are reward components for makespan and carbon emission, respectively. Therefore, the reward r_t at decision step t is defined according to Eq. 7:

$$r_t = w_1 r_t^{CT} + w_2 r_t^{CE} \tag{13}$$

To verify Eq. 13, the cumulative reward is calculated as:

$$\begin{aligned} R &= \sum_{t=1}^{T} r_t = \sum_{t=1}^{T}\left(w_1 r_t^{CT} + w_2 r_t^{CE}\right) \\ &= \sum_{t=1}^{T} w_1\left(CT(t) - CT(t+1)\right) + \sum_{t=1}^{T} w_2\left(CE(t) - CE(t+1)\right) \\ &= w_1\left(CT(1) - CT(T+1)\right) + w_2\left(CE(1) - CE(T+1)\right) \end{aligned} \tag{14}$$

where $CT(1)$ and $CE(1)$ are both zero as none of the operations is determined at the initial step. Once all the operations are determined after the T th decision step, i.e., $SO_i(T)$ is equal to n_i, $CT(T+1)$ and $CE(T+1)$ are equal to $C_{\max}$ and TCE, respectively. Therefore, Eq. 14 can be further simplified as:

$$R = -w_1 CT(T+1) - w_2 CE(T+1) = -\left(w_1 C_{\max} + w_2 TCE\right) \tag{15}$$

Eq. 15 indicates that maximizing the cumulative reward can reach the optimization objectives of minimizing the weighted sum of $C_{\max}$ and TCE.

3.2 Policy network

The goal of the scheduling policy is to determine the best-matched action for a given state. In this paper, a DNN with parameter θ consisting of six fully connected layers is employed to parameterize the scheduling policy denoted as $\pi_\theta(a_t|s_t)$. The input layer has ten neurons equal to the number of the state features, and the output layer outputs the probability distribution over the nine actions. Each of the first three hidden layers has sixty-four neurons while the fourth hidden layer has thirty-two neurons, and the Tanh activation function is used for all hidden neurons.

The PPO algorithm is adopted to train the policy network, where the state-value function $V(s_t)$ is approximated by another DNN with parameter $\varnothing$, denoted as $V_\varnothing(s_t)$. $V_\varnothing(s_t)$ has the same structure as $\pi_\theta(a_t|s_t)$ except that the output layer consists of only one neuron, and shares the first three hidden layers with $\pi_\theta(a_t|s_t)$ to utilize the learned abstract features.

3.3 Deep reinforcement learning training process

DRL establishes an interaction framework between the agent and the environment using the MDP components: state, action, and reward. The agent learns to optimize its decision-making policy through the interaction, i.e., tunning its policy network $\pi_\theta(a_t|s_t)$. Figure 3 illustrates the PPO-based DRL training process for the CEA-FJSP, where a training cycle includes a sampling phase and an update phase. Two same policy networks $\pi_{\theta_{old}}$ and π_θ are set-up in the beginning of training to facilitate the training process. During a training cycle, $\pi_{\theta_{old}}$ remains unchanged throughout the sampling and update phases, while π_θ is updated multiple times during the update phase.

In the sampling phase, $\pi_{\theta_{old}}$ interacts with the scheduling environment to collect sufficient state-action-reward tuples,

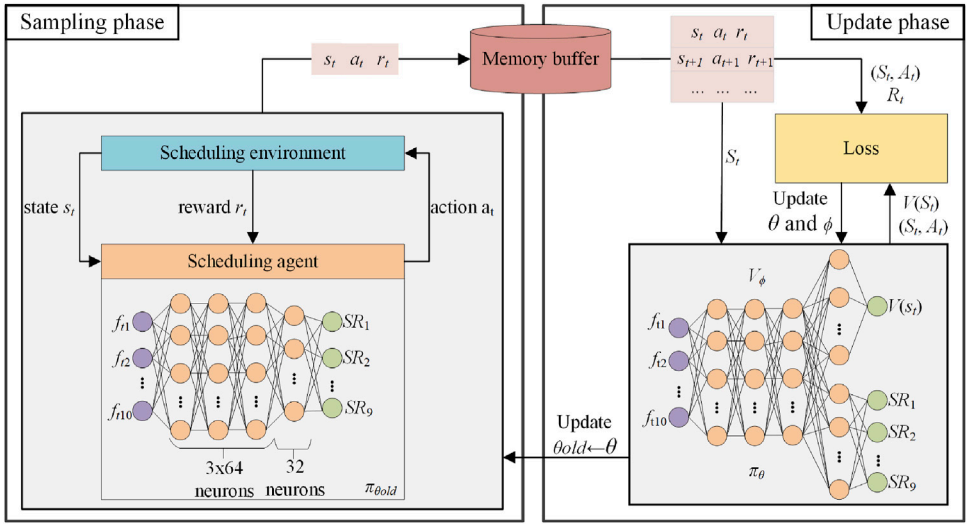

FIGURE 3
PPO-based DRL training process for CEA-FJSP.

(s_t, a_t, r_t), and store into a memory buffer. In the update phase, π_θ is updated for several epochs with the collected data. After that, $\pi_{\theta_{old}}$ copies π_θ and then starts the next training cycle. The surrogate objective loss function of the policy network is defined as:

$$L_t^{\mathrm{CLIP}} = \mathbb{E}_t\left[\min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t,\ \mathrm{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right)\right] \tag{16}$$

where $\mathbb{E}_t[\cdot]$ denotes the empirical average, $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the importance sampling weight, and $\mathrm{clip}(\cdot)$ is the constraint function with hyperparameter ε to ensure the similarity between π_θ and $\pi_{\theta_{old}}$. $\hat{A}_t$ is the generalized advantage estimation (GAE) function.

The value network is updated through the mean squared error (MSE) loss function:

$$L_t^{\mathrm{VF}} = \mathbb{E}_t\left[\left(V_\varnothing(s_t) - V_t^{targ}\right)^2\right] = \mathbb{E}_t\left[\left(V_\varnothing(s_t) - \sum_{i=t}^{T} r_i\right)^2\right] \tag{17}$$

Due to parameter sharing, the entire network model is trained with the loss function:

$$L_t^{\mathrm{CLIP+VF+S}} = \mathbb{E}_t\left[L_t^{\mathrm{CLIP}} - c_1 L_t^{\mathrm{VF}} + c_2 S[\pi_\theta](s_t)\right] \tag{18}$$

where $S[\pi_\theta](s_t)$ is the entropy bonus to encourage exploration, while c_1 and c_2 are the coefficients.

The pseudo-code of the training process is presented in Algorithm 1. Here, N training instances are initialized at the beginning of a training cycle to prevent the DRL scheduling model from overfitting a specific instance. The data collected from the sampling phase is used to calculate the cumulative gradients to update parameters θ and $\varnothing$ for K epochs.

Input: training cycles L; memory buffer M; update epochs K: number of training instances N
Output: π_θ
Initialize policy network π_θ and value network V_ϕ
Initialize old policy network $\pi_{\theta old}$
for *cycle* = 1, 2,..., L **do**
Randomly initialize N CEA-FJSP instances
for *instance* = 1, 2,..., N **do**
for *step* = 1, 2,..., T **do**
Randomly sample action a_t based on $\pi_{\theta old}$
Execute action a_t
Receive reward r_t
Transfer to the next state s_{t+1}
Store (s_t, a_t, r_t) in M
end for
end for
for *epoch* = 1, 2,..., K **do**
Compute L^{CLIP} by Eq. 16
Compute L^{VF} by Eq. 17
Compute $L^{CLIP+VF+S}$ by Eq. 18
Update parameter θ, ϕ with $\nabla L^{CLIP+VF+S}$
end for
$\pi_{\theta old} \leftarrow \pi_\theta$
end for

Algorithm 1. Training process for CEA-FJSP using PPO

TABLE 4 Brandimarte's benchmarks.

Configuration code	*n*	*m*	*nop*	*meq*	*proc*
Mk01	10	6	[5, 7]	3	[1, 7]
Mk02	10	6	[5, 7]	6	[1, 7]
Mk03	15	8	[10, 10]	5	[1, 20]
Mk04	15	8	[3, 10]	3	[1, 10]
Mk05	15	4	[5, 10]	2	[5, 10]
Mk06	10	15	[15, 15]	5	[1, 10]
Mk07	20	5	[5, 5]	5	[1, 20]
Mk08	20	10	[10, 15]	2	[5, 20]
Mk09	20	10	[10, 15]	5	[5, 20]
Mk10	20	15	[10, 15]	5	[5, 20]

TABLE 5 Parameters added to extend Brandimarte's benchmarks.

Parameter	Value
Operation processing time (s)	Unif (*proc*)
Machine processing power (kW)	Unif [4, 15]
Machine idle power (kW)	Unif [1, 2]
Coolant replacement cycle ($\times 10^4$s)	Rand (80, 85, 90, 95, 100)
Coolant recycling volume (L)	Rand (200, 250, 300, 350, 400)
Carbon emission factor α_e (kg/(kWh))	0.540
Carbon emission factor α_f (kg/L)	5.143

4 Experimental results and discussion

Four numerical experiments were conducted to train the DRL scheduling model, verify the optimization and generalization abilities, and explore the weight effect. The dataset used in the experiments was adapted from the benchmarks in Brandimarte (1993), referred as Brandimarte's benchmarks hereafter.

4.1 Experimental setting

4.1.1 Dataset adaption

Brandimarte's benchmarks defined some configurations for FJSP instances, as shown in Table 4. A benchmark is an FJSP instance consisting of n jobs and m machines, where a job has a range of nop operations, an operation can be processed by a range of meq candidate machines, and the processing time varies in the range denoted as $proc$.

Since the proposed CEA-FJSP considers energy consummation of machine operation and coolant treatment in addition to makespan. Therefore, Brandimarte's benchmarks are extended by adding seven additional parameters to generate CEA-FJSP scheduling instances. Table 5 lists the added parameters, where Unif denotes uniform distribution of real numbers and Rand denotes random selection. The processing time was measured in seconds instead of the unit time used in the original benchmarks to calculate the specific values of carbon emission. Carbon emission factors were set according to the Hong Kong SME Carbon Audit Toolkit (Liu et al., 2018). The Mk*i* instances of Brandimarte's benchmarks were changed to Mk*i*Ex instances after adding the additional parameters.

TABLE 6 Hyperparameter values of Algorithm 1.

Hyperparameter	Value
Trian cycles L	10000
Number of training instances N	5
Update epochs K	1
GAE coefficient λ	0.95
Discount factor γ	0.99
Clipping ratio ε	0.2
Value loss coefficient c_1	0.5
Entropy coefficient c_2	0.05
Learning rate α	0.0001
Optimizer	Adam

4.1.2 Evaluation metrics

Average makespan, AC, average total carbon emission, AC, and normalized performance, NP, were used to evaluate the performance of the proposed model. The smaller AC, AT or NP correspond to the better performance. These three metrics are defined as follows:

$$AC = \frac{1}{n}\sum_{i=1}^{n}(C_{\max})_i \tag{19}$$

$$AT = \frac{1}{n}\sum_{i=1}^{n}TCE_i \tag{20}$$

$$NP = w_1\frac{AC - \min\limits_{m_d\in \boldsymbol{MS}} AC_d}{\max\limits_{m_d\in \boldsymbol{MS}} AC_d - \min\limits_{m_d\in \boldsymbol{MS}} AC_d} + w_2\frac{AT - \min\limits_{m_d\in \boldsymbol{MS}} AT_d}{\max\limits_{m_d\in \boldsymbol{MS}} AT_d - \min\limits_{m_d\in \boldsymbol{MS}} AT_d} \tag{21}$$

where n is the total number of testing instances, $(C_{\max})_i$ and TCE_i are the makespan and total carbon emission of the i th instances. Method set, $\boldsymbol{MS}$, is composed of the proposed mode and the scheduling methods used for comparison, and d denotes the index of the scheduling method m_d. AC_d and AT_d denote AC and AT of m_d, respectively.

4.2 Training dynamics

Five Mk03Ex instances were generated in each training cycle based on the Mk03 configuration in Table 4 with the parameters

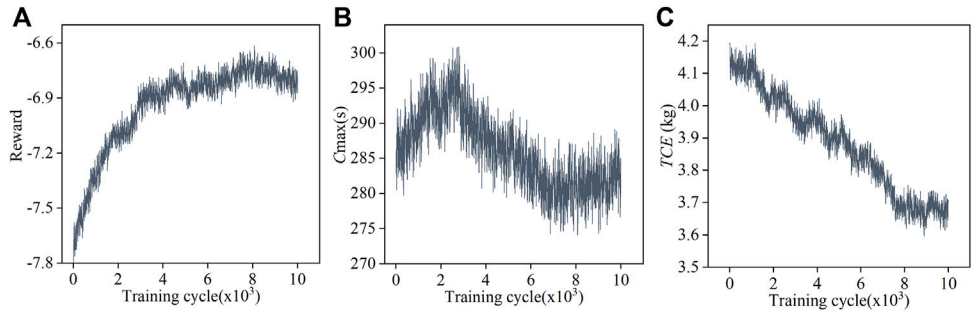

FIGURE 4
Training histories of **(A)** the reward, **(B)** the makespan, and **(C)** the total carbon emission.

in Table 5. These instances were used to train the proposed DRL scheduling model to produce the DRL-Mk03Ex scheduling solver. Table 6 lists the values of hyperparameters of Algorithm 1. Both weights w_1 and w_2 were set to 0.5 to equally evaluate the contribution of makespan and carbon emissions to the reward. Furthermore, reward scaling (Engstrom et al., 2020) was adopted to stabilize the training process. The hardware for training was a PC with a single Intel Xeon E5-2678 V3 @ 2.50 GHz CPU and a single NVIDIA RTX A2000 GPU. Algorithm 1 was implemented using Python 3.7, with PyTorch to deploy the network model.

Figure 4A–C show the training histories of the reward, $C_{\max}$ and *TCE*, respectively. It can be seen from Figure 4A that the reward gradually increases with the advance of the training process. Figure 4C shows that *TCE* continuously contributes positively to the reward, as it decreases monotonously along the timeline. It can be seen from Figure 4B that $C_{\max}$ increases till about the 2200th cycle and then decreases until the end. The results show that the contribution of *TCE* to the reward surpasses $C_{\max}$ in the early stage of optimization, and finally, both *TCE* and $C_{\max}$ are optimized by the DRL scheduling model. All three curves in Figure 4 begin to converge around the 7000th cycle and all of them oscillate slightly thereafter. Therefore, the training process had better stop around the 7000th cycle or the performance could get worse, exhibiting a kind of overfitting behavior.

4.3 Optimization ability

One hundred additional Mk03Ex instances different from those used in the training stage were generated to test the DRL-Mk03Ex against the proposed scheduling rules SR_1 to SR_9 and GA (Yin et al., 2017) respectively.

Figure 5 shows the performance of DRL-Mk03Ex over the Mk03Ex instances. It can be seen from the figure that DRL-Mk03Ex outperforms all the scheduling rules and GA on the testing instances, i.e., it achieves the lowest average makespan and the lowest average total carbon emission. Although GA and some scheduling rules (SR_5, SR_6, SR_8) perform well in reducing makespan or total carbon emission, none of the scheduling rules and GA can simultaneously minimize the two objectives. Furthermore, DRL-Mk03Ex is also significantly better than the scheduling rules and GA in terms of *NP*. The results confirm the superiority and optimization ability of DRL-Mk03Ex.

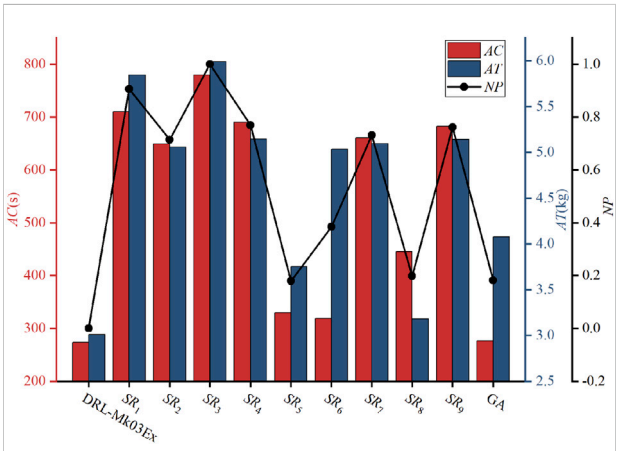

FIGURE 5
Performance of DRL-Mk03Ex over the Mk03Ex instances.

4.4 Generalization ability

The DRL scheduling solver built on the Mk03Ex instances (DRL-Mk03Ex) was tested on the Mk01Ex, Mk02Ex, and Mk04Ex to Mk10Ex instances. That is, to say, the instances used for testing were different from the ones used for training, and the difference was significant in the sense that the testing and

TABLE 7 Performance of DRL-Mk03Ex over the non-Mk03Ex instances.

Configuration code	Metric	DRL-Mk03Ex	SR_1	SR_2	SR_3	SR_4	SR_5	SR_6	SR_7	SR_8	SR_9	GA
MK01Ex	*AC* (s)	65.27	101.29	89.68	103.85	93.37	67.42	67.31	91.87	80.88	91.33	**52.67**
	AT (kg)	**0.52**	0.74	0.66	0.74	0.67	0.56	0.69	0.66	0.53	0.66	0.53
	NP	0.12	0.97	0.68	1.00	0.74	0.24	0.53	0.70	0.30	0.70	**0.02**
MK02Ex	*AC* (s)	52.86	100.50	90.38	103.75	90.12	62.33	63.93	91.10	62.62	93.86	**49.05**
	AT (kg)	0.44	0.82	0.68	0.83	0.69	0.53	0.76	0.69	**0.43**	0.69	0.55
	NP	**0.05**	0.96	0.69	1.00	0.70	0.25	0.55	0.71	0.12	0.73	0.15
MK04Ex	*AC* (s)	120.70	202.20	182.50	219.80	196.07	125.37	124.41	192.16	170.01	191.79	**98.07**
	AT (kg)	1.19	1.72	1.55	1.75	1.58	1.30	1.56	1.58	1.26	1.57	**1.03**
	NP	0.20	0.91	0.71	1.00	0.78	0.30	0.48	0.77	0.46	0.76	**0.00**
MK05Ex	*AC* (s)	281.31	413.55	387.96	420.45	392.61	274.41	**273.27**	393.11	385.33	393.70	280.55
	AT (kg)	2.49	2.29	2.50	2.29	**1.99**	2.34	2.29	2.13	2.29	2.49	2.15
	NP	0.52	0.77	0.89	0.79	0.41	0.35	0.29	0.54	0.67	0.90	**0.18**
MK06Ex	*AC* (s)	**166.06**	454.68	401.38	472.50	410.75	190.29	187.53	402.76	280.67	405.36	166.54
	AT (kg)	**1.99**	4.12	3.53	4.18	3.55	2.30	3.07	3.52	2.25	3.54	2.52
	NP	**0.00**	0.96	0.74	1.00	0.76	0.11	0.28	0.74	0.25	0.74	0.12
MK07Ex	*AC* (s)	**219.15**	412.82	372.82	444.81	391.11	274.53	277.09	382.54	278.85	376.05	219.85
	AT (kg)	**1.92**	3.44	2.95	3.48	2.99	2.42	3.26	2.98	1.93	2.98	2.53
	NP	**0.00**	0.92	0.67	1.00	0.72	0.28	0.56	0.70	0.14	0.69	0.20
MK08Ex	*AC* (s)	595.41	1314.55	1264.56	1388.23	1331.61	570.95	**567.96**	1301.90	1241.76	1292.80	632.53
	AT (kg)	**7.68**	10.43	9.93	10.62	10.09	7.76	8.50	9.99	9.29	9.99	7.78
	NP	**0.02**	0.92	0.81	1.00	0.88	**0.02**	0.14	0.84	0.68	0.83	0.06
MK09Ex	*AC* (s)	**530.05**	1380.95	1283.19	1475.75	1341.31	568.43	547.01	1304.44	1151.68	1297.93	550.78
	AT (kg)	**7.19**	11.62	10.45	11.86	10.59	7.63	9.46	10.50	8.67	10.49	8.34
	NP	**0.00**	0.92	0.75	1.00	0.79	0.07	0.25	0.76	0.49	0.76	0.13
MK10Ex	*AC* (s)	**516.53**	1354.37	1183.54	1421.84	1210.95	560.22	570.26	1185.87	1013.48	1195.13	521.93
	AT (kg)	**7.46**	13.63	11.70	13.89	11.80	8.00	10.57	11.70	9.13	11.73	8.88
	NP	**0.00**	0.94	0.70	1.00	0.72	0.07	0.27	0.70	0.40	0.71	0.11

The proposed scheduling rules SR^1 to SR^9 and GA are used as the baselines and the best values of each metric are highlighted in bold font.

the training instances were sampled from different configurations. To compare the performance, Table 7 shows the average results of three metrics over 100 instances for nine different instance configurations. The proposed scheduling rules SR_1 to SR_9 and GA are used as the baselines and the best values of each metric are highlighted in bold font in Table 7.

Table 7 shows that DRL-Mk03Ex achieves the best solutions in most instances compared with the scheduling rules and GA. Furthermore, Mk01Ex and Mk02Ex instances have a simpler configuration than the Mk03Ex instances, while Mk04Ex to Mk10Ex instances have a more complex configuration. This means the DRL-Mk03Ex can be bidirectionally generalized. Besides, DRL-Mk03Ex can achieve comparable performance with GA in the simple scheduling instances, while surpassing GA in the complex instances. Moreover, DRL-Mk03Ex is also more robust than the scheduling rules. For example, *AC* changes from 52.86 to 595.41 s for DRL-Mk03Ex, while from 100.5 to

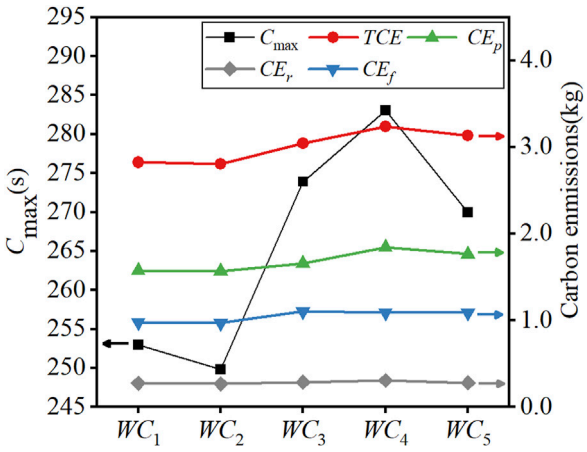

FIGURE 6
Effects of different weight combinations on makespan and carbon emission.

1380.95 s for SR_1. The performance fluctuates even more wildly for the complex instances among the scheduling rules. For example, SR_5 and SR_3 achieve 560.22 and 1421.84 s as AC values for the Mk10Ex instances, respectively.

4.5 Weight effect

The Mk03Ex instances were used to train the DRL scheduling model under various weight combinations (w_1, w_2): WC_1 = (0.0, 1.0), WC_2 = (0.25, 0.75), WC_3 = (0.5, 0.5), WC_4 = (0.75, 0.25), and WC_5 = (1.0, 0.0). Consequently, five DRL scheduling solvers were built. These solvers had the same model structure distinguished by different parameter values. Figure 6 illustrates the resultant $C_{\max}$ and TCE as well as carbon emission components, CE_p, CE_r and CE_f for processing state, idle state, and coolant treatment, respectively, of the five solvers.

The results demonstrate the nonlinearity of the DRL scheduling solvers. The $C_{\max}$ does not vary monotonously with the weight w_1; nor does the carbon emission with the weight w_2, i.e., either the makespan or the carbon emission is affected jointly by w_1 and w_2. It also implies that the DRL scheduling solvers cannot directly control the sub-optimization objectives. Instead, the weights should be treated as optimization parameters in the sense that the weighted optimization objective can be figured out for a given instance by adjusting w_1 and w_2. For example, WC_2 is the best among the five weight combinations for the Mk03Ex instances.

For each weight combination, the three carbon emission components, CE_p, CE_f, and CE_r, contribute roughly 56%, 34%, and 10% to TCE, respectively. Specifically, the machines produce the most carbon emission when processing an operation, and the least carbon emission when in the idle state. The carbon emission caused by the coolant treatment is also nonnegligible. Furthermore, it can be observed that as CE_p, CE_f, and CE_r maintain a similar tendency with the TCE along the weight change. The results stimulate the possibility to simplify the representation of carbon emission by replacing TCE with CE_p or with the sum of CE_p and CE_f.

5 Conclusion

In this study, a carbon emission-aware flexible job-shop scheduling problem denoted as CEA-FJSP is formulated and a DRL scheduling model is proposed to generate feasible scheduling solutions without extra searching. In the CEA-FJSP, the energy consumption of machine operation and the coolant treatment are identified as two main carbon emission sources. The proposed DRL scheduling model treats the CEA-FJSP as a Markov decision process where the scheduling agent interacts repeatedly with the scheduling environment, i.e., the temporary scheduling solution, to determine an appropriate action for a given state. The interaction is guided by the reward which represents the optimization objectives: minimizing makespan and carbon emission. The experimental results verify that the proposed DRL scheduling model achieves stronger optimization and generalization ability than the scheduling rules and GA, and the DRL scheduling model can be tuned by varying the weight combination. The future work should consider more carbon emission sources, more optimization objectives, and more flexible DRL framework to approach a more practical scheduling solution for the complex production scenarios.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

SW and JL were the principal authors for the text, and responsible for problem formulation, method design, experimental analysis, and manuscript writing. HT contributed to investigation. HT and JW contributed to the visualization together. All authors reviewed the final version of the manuscript and consented to publication.

Funding

This work was supported by the National Key R&D Program of China (Grant No. 2020YFB1708500), and the Science and Technology Planning Project of Guangzhou City (Grant No. 202102020882).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor UB declared a shared affiliation with the author HT at the time of review.

Publisher's note

References

Allahverdi, A., Ng, C. T., Cheng, T. E., and Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *Eur. J. Oper. Res.* 187 (3), 985–1032. doi:10.1016/j.ejor.2006.06.060

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* 34 (6), 26–38. doi:10.1109/MSP.2017.2743240

Bhatti, U. A., Nizamani, M. M., and Mengxing, H. (2022a). Climate change threatens Pakistan's snow leopards. *Science* 377 (6606), 585–586. doi:10.1126/science.add9065

Bhatti, U. A., Yan, Y., Zhou, M., Ali, S., Hussain, A., Qingsong, H., et al. (2021). Time series analysis and forecasting of air pollution particulate matter (PM 2.5): An SARIMA and factor analysis approach. *IEEE Access* 9, 41019–41031. doi:10.1109/ACCESS.2021.3060744

Bhatti, U. A., Zeeshan, Z., Nizamani, M. M., Bazai, S., Yu, Z., and Yuan, L. (2022b). Assessing the change of ambient air quality patterns in Jiangsu Province of China pre-to post-COVID-19. *Chemosphere* 288, 132569. doi:10.1016/j.chemosphere.2021.132569

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Ann. Oper. Res.* 41 (3), 157–183. doi:10.1007/BF02023073

Brucker, P., and Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. *Computing* 45 (4), 369–375. doi:10.1007/BF02238804

Dai, M., Tang, D., Giret, A., and Salido, M. A. (2019). Multi-objective optimization for energy-efficient flexible job-shop scheduling problem with transportation constraints. *Robot. Comput. Integr. Manuf.* 59, 143–157. doi:10.1016/j.rcim.2019.04.006

Du, Y., Li, J. Q., Chen, X. L., Duan, P. Y., and Pan, Q. K. (2022). Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job-shop scheduling problem. *IEEE Trans. Emerg. Top. Comput. Intell.* (Early Access), 1–15. doi:10.1109/TETCI.2022.3145706

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., et al. (2020). Implementation matters in deep policy gradients: A case study on PPO and trpo. *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2005.12729.

Feng, Y., Zhang, L., Yang, Z., Guo, Y., and Yang, D. (2021). "Flexible job-shop scheduling based on deep reinforcement learning," in 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT), Haikou, China, 29-31 October 2021 (IEEE), 660–666. doi:10.1109/ACAIT53529.2021.9731322

Fernandes, J. M., Homayouni, S. M., and Fontes, D. B. (2022). Energy-efficient scheduling in job shop manufacturing systems: A literature review. *Sustainability* 14 (10), 6264. doi:10.3390/su14106264

Gao, K., Huang, Y., Sadollah, A., and Wang, L. (2020). A review of energy-efficient scheduling in intelligent production systems. *Complex Intell. Syst.* 6 (2), 237–249. doi:10.1007/s40747-019-00122-6

Gutowski, T., Murphy, C., Allen, D., Bauer, D., Bras, B., Piwonka, T., et al. (2005). Environmentally benign manufacturing: Observations from Japan, europe and the United States. *J. Clean. Prod.* 13 (1), 1–17. doi:10.1016/j.jclepro.2003.10.004

Han, B. -A., and Yang, J. -J. (2020). Research on adaptive job-shop scheduling problems based on dueling double DQN. *IEEE Access* 8, 186474–186495. doi:10.1109/ACCESS.2020.3029868

Han, B. A., and Yang, J. J. (2021). A deep reinforcement learning based solution for flexible job-shop scheduling problem. *Int. J. Simul. Model.* 20 2, 375–386. doi:10.2507/IJSIMM20-2-CO7

Lang, S., Behrendt, F., Lanzerath, N., Reggelin, T., and Müller, M. (2020). "Integration of deep reinforcement learning and discrete-event simulation for real-time scheduling of a flexible job shop production," in 2020 Winter Simulation Conference (WSC), Orlando, FL, USA, 14-18 December 2020 (IEEE), 3057–3068. doi:10.1109/WSC48552.2020.9383997

Lei, D., Zheng, Y., and Guo, X. (2017). A shuffled frog-leaping algorithm for flexible job-shop scheduling with the consideration of energy consumption. *Int. J. Prod. Res.* 55 (11), 3126–3140. doi:10.1080/00207543.2016.1262082

Li, M., and Wang, G. G. (2022). A review of green shop scheduling problem. *Inf. Sci. (N. Y).* 589, 478–496. doi:10.1016/j.ins.2021.12.122

Liu, C. -L., Chang, C. -C., and Tseng, C. -J. (2020). Actor-critic deep reinforcement learning for solving job-shop scheduling problems. *IEEE Access* 8, 71752–71762. doi:10.1109/ACCESS.2020.2987820

Liu, Q., Tian, Y., Wang, C., Chekem, F. O., and Sutherland, J. W. (2018). Flexible job-shop scheduling for reduced manufacturing carbon footprint. *J. Manuf. Sci. Eng.* 140 (6), 061006. doi:10.1115/1.4037710

Liu, R., Piplani, R., and Toro, C. (2022). Deep reinforcement learning for dynamic scheduling of a flexible job shop. *Int. J. Prod. Res.* 60 (13), 4049–4069. doi:10.1080/00207543.2022.2058432

Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Appl. Soft Comput.* 91, 106208. doi:10.1016/j.asoc.2020.106208

Luo, S., Zhang, L., and Fan, Y. (2021). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Comput. Ind. Eng.* 159, 107489. doi:10.1016/j.cie.2021.107489

Luo, S., Zhang, L., and Fan, Y. (2019). Energy-efficient scheduling for multi-objective flexible job shops with variable processing speeds by grey wolf optimization. *J. Clean. Prod.* 234, 1365–1384. doi:10.1016/j.jclepro.2019.06.151

Mokhtari, H., and Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Comput. Chem. Eng.* 104, 339–352. doi:10.1016/j.compchemeng.2017.05.004

Monaci, M., Agasucci, V., and Grani, G. (2021). An actor-critic algorithm with deep double recurrent agents to solve the job-shop scheduling problem. *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2110.09076.

Naimi, R., Nouiri, M., and Cardin, O. (2021). A Q-Learning rescheduling approach to the flexible job shop problem combining energy and productivity objectives. *Sustainability* 13 (23), 13016. doi:10.3390/su132313016

Ni, F., Hao, J., Lu, J., Tong, X., Yuan, M., Duan, J., et al. (2021). "A multi-graph attributed reinforcement learning based optimization algorithm for large-scale hybrid flow shop scheduling problem," in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Singapore, Aug 14, 2021 - Aug 18, 2021, 3441–3451. doi:10.1109/ICCECE54139.2022.9712705

Pan, Z., Wang, L., Wang, J., and Lu, J. (2021). Deep reinforcement learning based optimization algorithm for permutation flow-shop scheduling. *IEEE Trans. Emerg. Top. Comput. Intell.* (Early Access), 1–12. doi:10.1109/TETCI.2021.3098354

Park, J., Chun, J., Kim, S. H., Kim, Y., and Park, J. (2021). Learning to schedule job-shop problems: Representation and policy learning using graph neural network and reinforcement learning. *Int. J. Prod. Res.* 59 (11), 3360–3377. doi:10.1080/00207543.2020.1870013

Qu, S., Wang, J., and Shivani, G. (2016). "Learning adaptive dispatching rules for a manufacturing process system by using reinforcement learning approach," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 06-09 September 2016 (IEEE). doi:10.1109/ETFA.2016.7733712

Ren, J. F., Ye, C. M., and Yang, F. (2020). A novel solution to JSPS based on long short-term memory and policy gradient algorithm. *Int. J. Simul. Model.* 19 (1), 157–168. doi:10.2507/IJSIMM19-1-CO4

van Ekeris, T., Meyes, R., and Meisen, T. (2021). "Discovering heuristics and metaheuristics for job-shop scheduling from scratch via deep reinforcement learning," in Proceedings of the Conference on Production Systems and Logistics (CPSL), Online, 10-11 August 2021, 709–718. doi:10.15488/11231

Wu, X., and Sun, Y. (2018). A green scheduling algorithm for flexible job shop with energy-saving measures. *J. Clean. Prod.* 172, 3249–3264. doi:10.1016/j.jclepro.2017.10.342

Xu, B., Mei, Y., Wang, Y., Ji, Z., and Zhang, M. (2021). Genetic programming with delayed routing for multiobjective dynamic flexible job-shop scheduling. *Evol. Comput.* 29 (1), 75–105. doi:10.1162/evco_a_00273

Yan, Q., Wu, W., and Wang, H. (2022). Deep reinforcement learning for distributed flow shop scheduling with flexible maintenance. *Machines* 10 (3), 210. doi:10.3390/machines10030210

Yin, L., Li, X., Gao, L., Lu, C., and Zhang, Z. (2017). A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem. *Sustain. Comput. Inf. Syst.* 13, 15–30. doi:10.1016/j.suscom.2016.11.002

Zeng, Y., Liao, Z., Dai, Y., Wang, R., and Yuan, B. (2022). Hybrid intelligence for dynamic job-shop scheduling with deep reinforcement learning and attention mechanism. *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2201.00548.

Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P. S., and Xu, C. (2020). Learning to dispatch for job-shop scheduling via deep reinforcement learning. *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2010.12367.

Zhang, H., Xu, G., Pan, R., and Ge, H. (2022). A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time. *Eng. Optim.* 54 (10), 1646–1667. doi:10.1080/0305215X.2021.1949007

Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *J. Intell. Manuf.* 30 (4), 1809–1830. doi:10.1007/s10845-017-1350-2

Zhao, Y., Wang, Y., Tan, Y., Zhang, J., and Yu, H. (2021). Dynamic jobshop scheduling algorithm based on deep q network. *IEEE Access* 9, 122995–123011. doi:10.1109/ACCESS.2021.3110242