Robotics and Computer–Integrated Manufacturing 80 (2023) 102454

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

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

Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning

Yongkui Liu [a,*], Yaoyao Ping [a], Lin Zhang [b], Lihui Wang [c], Xun Xu [d]

[a] School of Mechano Electronic Engineering Xidian University, Xi'an, Shaanxi 710071, China
[b] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[c] Department of Production Engineering KTH Royal Institute of Technology, Stockholm 10044, Sweden
[d] Department of Mechanical Engineering, The University of Auckland, Auckland 1142, New Zealand

ARTICLE INFO

Keywords:
Cloud manufacturing
Scheduling
Robot service
Deep reinforcement learning
Dueling DQN

ABSTRACT

Cloud manufacturing is a service-oriented manufacturing model that offers manufacturing resources as cloud services. Robots are an important type of manufacturing resources. In cloud manufacturng, large-scale distributed robots are encapsulated into cloud services and provided to consumers in an on-demand manner. How to effectively and efficiently manage and schedule decentralized robot services in cloud manufacturing to achieve on-demand provisioning is a challenging issue. During the past few years, Deep Reinforcement Learning (DRL) has become very popular and successfully been applied to many different areas such as games, robotics, and manufacturing. DRL also holds tremendous potential for solving scheduling issues in cloud manufacturing. To this end, this paper is devoted to exploring effective approaches for scheduling of decentralized robot manufacturing services in cloud manufacturing with DRL. Specifically, both Deep Q-Networks (DQN) and Dueling Deep Q-Networks (DDQN)-based scheduling algorithms are proposed. Performance of different algorithms, including DQN, DDQN, and other three benchmark algorithms, indicates that DDQN performs the best with respect to each indicator. Effects of different combinations of weight coefficients and influencing degrees of different indicators on the overall scheduling objective are analyzed. Results indicate that the DDQN-based scheduling algorithm is able to generate scheduling solutions efficiently.

Notations

a_t	action taken by an agent at time t
ap_i	arm parameter of R_i
At_k	arrival time of O_k
$A^{\pi}(s,a)$	value function of action a in state s
b	minimum batch size
bp_i	base parameter of R_i
C	service users set
C_k	service user k
ep_i	excellent product ratio of R_i
$Et_{i,k}$	total execution time for R_i to complete O_k
f_i	functional type of R_i
F	number of functional type
Ft_k	functional type of O_k
gp_i	gear parameter of R_i
I	number of robot services
J	number of logistics services
K	number of service users
l	learning rate
loc_i	geographical location of R_i
loc_j^l	geographical location of L_j^l
$loc_{l_i}^l$	geographical location of l_i
Loc_k	geographical location of C_k
lp_i	life span index of R_i
L	logistics service pool
L_j^l	logistics service j
$L_{l_i}^l$	logistics service of $R_i (l_i < J)$
Lp_{i,l_i}^l	logistics price between $L_{l_i}^l$ and R_i
$Lp_{i,k}^l$	logistics price between R_i and C_k
L_j^l	logistics service j
Lp_{i,l_i}^l	logistics price between $L_{l_i}^l$ and R_i

* Corresponding author.
E-mail addresses: yongkuiliu@163.com (Y. Liu), lihuiw@kth.se (L. Wang), x.xu@auckland.ac.nz (X. Xu).

https://doi.org/10.1016/j.rcim.2022.102454
Received 31 January 2022; Received in revised form 10 August 2022; Accepted 15 September 2022
Available online 25 September 2022

$Lp^l_{i,k}$	logistics price between R_i and C_k
$Lp^l_{i,l_i,k}$	total logistics price between $L^l_{l_i}$ and C_k
Lt^l_{i,l_i}	logistics time between $L^l_{l_i}$ and R_i
$Lt^l_{i,k}$	logistics time between R_i and C_k
$Lt^l_{i,l_i,k}$	total logistics time between $L^l_{l_i}$ and C_k
$Maxs$	set of robot service life period
n_i	number of tasks executed in R_i
n_k	number of candidate service of O_k
$nlp_{i,k}$	normalized $lp_{i,k}$
nMs_{i,l_ik}	normalized Ms_{i,l_ik}
$nqg_{i,k}$	normalized $qg_{i,k}$
$nrel_{i,k}$	normalized $rel_{i,k}$
$nsf_{i,k}$	normalized $sf_{i,k}$
$nSp^l_{i,l_i,k}$	normalized $Sp^l_{i,l_i,k}$
$nTp_{i,l_i,k}$	normalized $Tp_{i,l_i,k}$
N	capacity of experience replay buffer
N_k	quantity of O_k
p_i	price of R_i for completing a unit amount of task for unit time
p^l_j	price of transporting unit weight of parts/blanks for unit distance of L^l_j
RP	a multi-objective function
RP_i	performance indicator set of R_i
RP^l_j	performance indicator set of L^l_j
$RP^l_{l_i}$	performance of $L^l_{l_i}$
RQ	multi-objective function
RQ_i	quality indicator set of R_i
RQ^l_j	quality indicator set of L^l_j
$RQ^l_{l_i}$	quality of $L^l_{l_i}$
R	robot service pool
R_{k,n_k}	$n_k th$ candidate service of O_k
R_i	robot service i
RQ_i	qualify of R_i
qg_i	quality grade of R_i
$Q^\pi(s,a)$	network output value
rel_i	reliability of R_i
$rel^l_{i,l_i,k}$	reliability for $L^l_{l_i}$ to transport O_k
rel^l_j	reliability of L^l_j
$Rp_{i,k}$	robot price for R_i to complete O_k
s_t	state of environment at time t
$sl^l_{i,l_i,k}$	security level of $L^l_{l_i}$ selected to transport O_k
sl^l_j	security level of L^l_j
Sp^l_j	risk probability of L^l_j
$Sp^l_{i,l_i,k}$	risk probability for R_i to complete O_k
t_i	time for R_i to complete a unit amount of task
sf_i	specifications of R_i
O_k	requirement of C_k
$Tp_{i,l_i,k}$	total price for R_i
Tp^l_j	total price index of L^l_j
$Tt_{i,k}$	total completion time for R_i to complete
Tt_i	total completion time index of R_i
Tt^l_j	total completion time index of L^l_j
w_a	weighting coefficients of ap_i
w_e	weighting coefficients of ep_i
w_b	weighting coefficients of bp_i
w_g	weighting coefficients of gp_i
w_q	weighting coefficients of qp_i
Wt_k	waiting time of O_k
$Wt_{i,k}$	total waiting time for R_i to complete O_k
Wt_{k,n_k}	waiting time of O_k for the $n_k th$ candidate service
WT_k	unit weight of O_k
y_{hig}	highest value of lp_i range
y_{low}	lowest value of lp_i range
u_{max}	maximum value of corresponding quality indicator
u_{min}	minimum value of corresponding quality indicator
α	parameter for full connection layer
β	parameter for full connection layer
θ	parameter of the convolution layer
γ	discounted factor
ε	maximal exploration value
$\lambda^l_{l_i}$	security coefficient of $L^l_{l_i}$
λ^l_j	security coefficient of L^l_j
$\|A\|$	number of a discrete action set

1. Introduction

Cloud manufacturing is a manufacturing model that encapsulates distributed manufacturing resources into cloud services and provide them to consumers over the Internet in an on-demand manner. Industrial robots are an important type of manufacturing resources in cloud manufacturing, which can perform various types of operations such as machining, assembly, welding, stacking, ect. In cloud manufacturing, geographically distributed industrial robots provided by different enterprises are transformed into robot cloud services and offered to consumers according to their requirements. There are massive robot cloud services in a cloud manufacturing system, and they have different functions and performance. Furthermore, scheduling processes in cloud manufacturing are usually accompanied by wide-area logistics. In this context, how to effectively and efficiently manage scheduling processes of large-scale robot services in cloud manufacturing is a challenging issue.

Scheduling is one of the most intensively studied topics in the area of cloud manufacturing, and various algorithms have been employed to solve this issue, including genetic algorithms, simulated annealing, tabu search, ant colony optimization, particle swarm optimization, and artificial bee colonies [1–4]. However, those algorithms are troublesome or perform poorly in solving scheduling issues in cloud manufacturing, reflected by the fact that they need to be redesigned for reasons such as changes of cloud environments and scheduling objectives. Consequently, cloud manufacturing calls for new and effective scheduling algorithms.

During the past few years, Deep Reinforcement Learning (DRL) that combines deep neural networks (DNN) with reinforcement learning (RL) has been successfully applied to many different areas such as games, robotics, computer vision, health, transportation, and manufacturing [5,6]. In particular, DRL has recently been applied to cloud manufacturing [7], and demonstrated its powerful capability in overcoming the limitations with the above-mentioned traditional algorithms. In view of this, this paper is devoted to exploring effective approaches for scheduling of decentralized robot manufacturing services in cloud manufacturing with DRL. Both Deep Q-Networks (DQN)- and Dueling Deep Q-Networks (DDQN)-based scheduling algorithms are proposed. Performance of different algorithms, including DQN, DDQN, and other three benchmark algorithms, i.e., random scheduling, round-robin scheduling, earliest scheduling, indicate that DDQN performs the best with respect to each indicator. Effects of different combinations of weight coefficients and influencing degrees of different indicators on the overall scheduling objective are analyzed. Results indicate that the DDQN-based scheduling algorithm is able to generate scheduling solutions.

The main contributions of this paper are as follows. Two DRL-based scheduling algorithms, including a DQN-based algorithm and a DDQN-based one for scheduling of decentralized robot services, are proposed. Regarding robot services, both robot quality (RQ) such as life span, specifications, quality grade, and reliability and performance of robot services (RP) (e.g., total price index and total completion time index) are considered during modeling. Regarding the research contents, apart

from algorithm performance comparison, effects of different combinations of weighting coefficients and the influence degrees of different indicators are also investigated. The results are interesting, indicating that the DDQN-based algorithm outperforms the DQN-based one and other benchmark algorithms, and the reliability of logistics service and execution time are the most influential indicators.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the problem description, related model, and objective analysis. Section 4 describes a DRL-based scheduling of decentralized robot services with DDQN algorithm. In Section 5, a case study is presented, and corresponding results are given. Finally, conclusions and future work are discussed in Section 6.

2. Literature review

This section briefly reviews work on topics of cloud manufacturing scheduling, DRL, and DRL-based manufacturing scheduling.

2.1. Scheduling in cloud manufacturing

Various methods have been proposed to solve scheduling issues in cloud manufacturing, including exact approaches, game theory-based approaches, bio-inspired metaheuristics, and machine learning and artificial intelligence [8]. Akbaripour et al. [4] proposed a mixed-integer programming models for solving the service selection optimization and scheduling problem with basic composition structures (i.e., sequential, parallel, loop, and selective). Liu et al. [9] studied workload-based multi-task scheduling with Monte Carlo methods and results indicated that scheduling larger workload tasks with a higher priority can shorten the makespan and increase service utilization. Meanwhile, Liu et al. [10, 11] studied multi-agent-based scheduling in cloud manufacturing and proposed a platform-level scheduling multi-agent system (MAS) and an enterprise level scheduling MAS. Zhou et al. [12] analyzed a dynamic task scheduling process and presented a method of dynamic scheduling based on real-time simulation. Moreover, Zhou et al. [12–15] proposed a simulation-based method to deal with task scheduling problems in a dynamic cloud manufacturing environment. Li et al. [16–18] proposed an ant colony optimization-based multi-objective algorithm and NSGA-II-based multi-objective algorithm to solve the multi-task scheduling problem, and a two scheduling strategies based on the two-level multi-task scheduling model were presented and evaluated. Liu et al. [19] proposed a 3D printing service scheduling method to decrease the delivery time of tasks from service suppliers to service demanders and generated optimal service scheduling solutions. Hu et al. [20] proposed the chaos optimization algorithm to solve the objective function and achieved scheduling of manufacturers under different scheduling tasks.

During past few years, robotics applications have been built around the computing paradigms of cloud computing and service oriented architecture [21,22]. At the same time, there are some studies around scheduling of robot resources, robotic product customization, and capability assessment of industrial robots in cloud manufacturing [23, 24]. Li et al. [3] proposed a cloud manufacturing scheduling model for efficiently exploiting distributed robot resources that can cooperatively handle a batch of tasks. Zhao et al. [25] considered a unified sustainable manufacturing capability of the industrial robots model in terms of functional attributes, structural information, activities and process condition in cloud manufacturing. Zhang et al. [26] developed a framework for designing a cloud-based ubiquitous robotic system which consists of the function, structure and behavior. Wang et al. [27] built cloud-based services of monitoring, process planning, machining and assembly in cloud manufacturing and considered a case study of remote control of a robotic assembly cell. Zhang et al. [28] considered a set of indicators of industrial robots in the context of cloud manufacturing and proposed a dynamic manufacturing capability assessment of industrial robots based on feedback information in cloud manufacturing. Yang et al. [29] proposed a cloud-edge-device collaboration framework of cloud manufacturing to support smart collaborative decision-making for smart robots. Zhao et al. [30] proposed a development framework for a digital twin industrial robot production line with closed-loop control based on a mechatronics approach. Wang et al. [31] proposed an improved hybrid optimization algorithm which integrates Gini impurity of Extreme Gradient Boosting model into Particle Swarm Optimization to solve the low performance and low efficiency problem of embedded feature selection method.

2.2. DRL

DRL combines the perception ability of deep learning (DL) and the decision-making ability of RL. It is an artificial intelligence method that is closer to the way of human thinking. DRL allows learning of control policies directly from high-dimensional sensory input using RL. Seven Atari 2600 games from the Arcade Learning Environment were tested via DRL, and results indicated that it outperformed all of previous approaches [32]. In order to solve overestimation problems under certain conditions, Double Q-learning algorithm was introduced in a tabular setting to reduce the observed overestimations [33]. In addition, a new DNN architecture for model-free RL - dueling network - was proposed and showed the state value function and the state-dependent action advantage function to obtain better policy evaluation in the presence of many similar-valued actions [34]. The results above depend on the proposed prioritized experience replay, so as to replay important transitions more frequently, and therefore learn more efficiently [35,36]. The above methods are value-based algorithms that are suited for solving discrete action decision problems. For continuous action decision problems, policy gradient algorithms are suitable and needed. An actor-critic, model-free algorithm based on the deterministic policy gradient was presented to solve the issues of continuous action spaces such as cartpole swing-up and dexterous manipulation [37,38].

2.3. DRL-based manufacturing scheduling

DRL has already been used in the area of cloud manufacturing service composition and scheduling by some researchers. Liang et al. [7] proposed a DDQN with prioritized replay named PD-DQN as the DRL algorithm to effectively address the issue of cloud manufacturing service composition. Liu et al. [39] proposed a DRL-based framework for scheduling in cloud manufacturing and presented a DRL model for online single-task scheduling in cloud manufacturing. Zhou et al. [40] analyzed the smart manufacturing service scheduling problem and proposed a DRL-based method to minimize the maximum completion time of all tasks. Zhu et al. [41] proposed a DRL-based method that converts scheduling problems with multiple resources into one learning target and learns effective strategies automatically. Yang et al. [42] proposed a DRL system of scheduling and reconfiguration to minimize the total tardiness cost in smart manufacturing. Du et al. [43] proposed a collaborative optimization method of service scheduling based on DRL to realize a comprehensive performance improvement of the whole manufacturing system. Mei et al. [44] studied the application of DRL in the multi-robotic disassembly line balance problem to minimize workstation idle time, priority disassembly of high-demand components and energy consumption. Yin et al. [45] presented a decentralized framework of multi-task allocation with attention (MTAA), and proposed a MTAA-DRL method to achieve task assignment equilibrium.

From the literature review above we can learn that scheduling in cloud manufacturing has received much attention and various intelligent optimization algorithms have been proposed. As a new technique, DRL has been used for solving service scheduling issues in cloud manufacturing, but related research is still in its infancy. Industrial robots are an important type of manufacturing resources in cloud manufacturing, but related studies including scheduling of robot services are not enough. In particular, scheduling of decentralized robot services using DRL has rarely been touched.

3. Problem formulation

We consider scheduling of decentralized robot services involved in a cloud manufacturing platform that either offers many different types of manufacturing resources (including robots, i.e. a universal cloud platform) or focuses exclusively on robot resources (i.e. a robot cloud service platform). Providers who own various types of industrial robots publish their resources to the cloud platform for different consumers to subscribe. Under the unified management and scheduling of the cloud operator, robot service requirements of consumers can be satisfied in an on-demand manner. The process of scheduling is accompanied by wide-area logistics between consumers and providers. The research is based on the following assumptions:

(1) Time constraints on tasks' total completion time are not considered.
(2) A robot service cannot be interrupted when it is working on a task.
(3) A robot service can be occupied by only one task at a time.

3.1. Robot and logistics services

Robot services and logistics services in the cloud platform are denoted by $R = \{R_i | 1 \le i \le I\}$ and $L = \{L_j^l \big| 1 \le j \le J\}$, respectively, where I and J represents, respectively, the total numbers of robot services and logistics services. R_i is the ith robot services, and can be described by Eq. (1).

$$R_i = \langle RQ_i, RP_i \rangle \tag{1}$$

where RQ_i is the set of quality attributes of robots, and RP_i denotes performance attributes of robot services. Specifically,

- $RQ_i = \{lp_i, tp_i, qg_i, rel_i\}$ $(1 \le i \le I)$, where lp_i, sf_i, qg_i, and rel_i represents, respectively, the life span, specifications, quality grade, and reliability of R_i;
- $RP_i = \{Tp_i, Tt_i\}$ $(1 \le i \le I)$, where Tp_i and Tt_i represents, respectively, the total price and total completion time of R_i.

Similarly, L_j^l can be expressed as follows:

$$L_j^l = \left\langle LQ_j^l, LP_j^l \right\rangle \tag{2}$$

where LQ_j^l and LP_j^l represent quality attributes and performance attributes of logistics services, respectively. Specifically,

- $LQ_j^l = \{rel_j^l\}$ $(1 \le j \le J)$, where rel_j^l represents the reliability of L_j^l;
- $LP_j^l = \{Sp_j^l, Tp_j^l, Tt_j^l\}$ $(1 \le j \le J)$, where Sp_j^l, Tp_j^l and Tt_j^l represent, respectively, the risk probability, the total price, and the total completion time of L_j^l.

Robot services can be described in more detail:

$$R_i = \langle Tp_i, Tt_i, lp_i, sf_i, qg_i, rel_i \rangle \tag{3}$$

where

- $Tp_i = \{loc_i, p_i\}$ $(1 \le i \le I)$, and loc_i and p_i are, respectively, the geographical location and price of R_i for using a unit amount of service for a unit time;
- $Tt_i = \{t_i, f_i\}$ $(1 \le i \le I)$ where t_i represents the time of R_i for completing a unit amount of task, and f_i $(1 \le f_i \le F)$ represents the functional type of R_i wth F being the total number of functional types of robot services in the entire cloud manufacturing system;
- $lp_i = \{lp_i | (1 \le i \le I)\}$ where lp_i represents the life span of R_i;
- $tp_i = \{bp_i, ap_i, gp_i\}$ $(1 \le i \le I)$ where bp_i, ap_i, and gp_i are the base parameter, arm parameter, and gear parameter, respectively;
- $qg_i = \{ep_i, qp_i\}$ $(1 \le i \le I)$ where ep_i and qp_i are, respectively, the excellent product ratio and qualified product ratio of P_i;
- $rel_i = \{rel_i | (1 \le i \le I)\}$ where rel_i represents the reliability of R_i.

Logistics services can also be described in more detail as follows:

$$L_j^l = \left\langle Tp_j^l, Tt_j^l, Sp_j^l, rel_j^l \right\rangle \tag{4}$$

where

- $Tp_j^l = \{p_j^l \big| (1 \le j \le J)\}$, where p_j^l is the price of L_j^l of transporting a unit weight of parts for a unit distance;
- $Tt_j^l = \{loc_j^l, v_j^l\}$ $(1 \le j \le J)$, where loc_j^l and v_j^l are, respectively, the geographical location and average transport speed of L_j^l;
- $Sp_j^l = \{sl_j^l, \lambda_j^l\}$ $(1 \le j \le J)$, where sl_j^l and λ_j^l represent, respectively, the security level and security coefficient of L_j^l;
- $rel_j^l = \{rel_j^l \big| (1 \le j \le J)\}$, where rel_j^l represents the reliability of L_j^l.

3.2. Consumers, orders and tasks

The set of consumers in the cloud manufacturing system is represented by $C = \{C_k | (1 \le k \le K)\}$, where K is the total number of consumers in the cloud manufacturing system. The kth consumer can be expressed as:

$$C_k = \{Loc_k, O_k\} \tag{5}$$

where Loc_k and O_k represent, respectively, the geographical location of C_k and his or her order.

O_k can be described as:

$$O_k = \{At_k, Wt_k, \ N_k, Ft_k, WT_k\} \tag{6}$$

where At_k, Wt_k, N_k, Ft_k, and WT_k represent the arrival time and waiting time of O_k, the number of tasks in O_k, the functional type of required robot service of all tasks in O_k, and the unit weight of a task in O_k, respectively. It should be noted that all of the tasks in O_k are completely identical, and can be completed by a single robot service.

3.3. Indexes and objectives

Both quality attributes and performance attributes for both robots and logistics are considered in calculating the scheduling objective. The former includes RQ and LQ, and the latter RP and LP.

3.3.1. RQ and LQ

(1) ***RQ***

RQ indexes consist of life span [46], specifications, quality grade, and reliability. The value of lp_i is sampled randomly from a certain probability distribution.

$$Maxs = \left\{ lp_i | lp_i \in \left[y_{low}, y_{high} \right] \right\} \tag{7}$$

where $Maxs$, y_{low}, *and* y_{high} are, respectively, the set of lifespans of robot services, and the lower and upper values of lp_i.

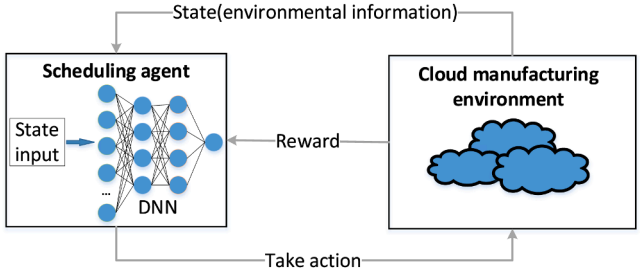

Fig. 1. Schematic diagram of DRL-based scheduling in cloud manufacturing.

Inspired by [47], specifications and quality grade can be computed as follows. The former is directly related to parameters of a robot such as its base parameter, arm parameter, and gear parameter, and can be computed according to the following formula.

$$sf_i = w_b \times bp_i + w_a \times ap_i + w_g \times gp_i \tag{8}$$

where w_b, w_a, and w_g are, respectively, the weighting coefficients of bp_i, ap_i, and gp_i with $w_b + w_a + w_g = 1$.

Similarly, quality grade can be computed according to the following formula.

$$qg_i = w_e \times ep_i + w_q \times qp_i \tag{9}$$

where w_e and w_q are, respectively, weighting coefficients of ep_i, qp_i, and $w_e + w_q = 1$.

The reliability for R_i to complete O_k can be denoted as $rel_{i,k}$.

(2) LQ

Assume that the logistics service for R_i is $L^l_{l_i}(l_i < J)$, then the reliability for $L^l_{l_i}$ to transport O_k can be denoted by $rel^l_{i,l_i,k}$.

3.3.2. RP and LP

RP consists of the price and time of a robot service, and LP consists of the price, time and risk probability of a logistics service.

(1) Total completion time

Total completion time includes execution time, waiting time, and logistics time. Execution time $Et_{i,k}$ for R_i to complete O_k can be computed by Eq. (10).

Table 1
Pseudo-code of the DDQN-based scheduling algorithm of decentralized robot services.

Algorithm DRL-based scheduling of decentralized robot services with DDQN algorithm	
1:	Initialize learning rate l, minibatch size b, discounted factor γ, maximal exploration value ε
2:	Initialize replay memory D with capacity N
3:	Initialize Q-network θ, α, β and target-network parameters $\theta^-, \alpha^-, \beta^-$
4:	for each episode do
5:	for each step t (task O_k) do
6:	Reset cloud manufacturing environment to initial state $s = s_t$
7:	With probability ε select a random action (service) $a = a_t$ Otherwise select $a_t = argmax_a Q(s_t, a_t; \theta, \alpha, \beta)$
8:	Schedule action a_t, observe reward r_t and next state (next order O_{k+1}) $s' = s_{t+1}, s_t = s_{t+1}$
9:	Store transition $(s_t,\ a_t,\ r_t, s_{t+1})$ in D
10	if episode terminates at step $j+1$ then
11:	set $y_j = r_j$
14:	else
15:	$y_j = r_j + \gamma max_{a'} Q(s_{j+1}, a'; \theta^-, \alpha^-, \beta^-)$
16:	end if
17:	Update Q-network parameters $\theta, \alpha,$ *and* β with a loss function of $L(\theta, \alpha, \beta) = \frac{1}{N}[(y_j - Q(s, a; \theta, \alpha, \beta))^2]$
18:	Compute TD-error $\delta_j = y_j - Q(s_t, a_t; \theta, \alpha, \beta)$
19	Every C steps reset $\theta^-, \alpha^-, \beta^- \leftarrow \theta, \alpha, \beta$
20:	end for
21:	end for

Table 2
Model variables and parameters.

Parameters	Value	Unit	Parameters	Value	Unit
I	100		ep_i	[0.8,1)	
J	5		qp_i	(0,0.2)	
F	10		loc^l_j	[0,500]	
loc_i	[0,500]	km	v^l_j	[80,100]	h/km
t_i	[5,10]		p^l_j	[0.05,0.2]	Yuan(kg/km)
p_i	[30,90]	Yuan	sl^l_j	[0.1,1)	
lp_i	[10,15]	year	λ^l_j	[0.1,1)	
bp_i	[30,90]		rel^l_j	[0.8,1)	
ap_i	[30,90]		w_q	0.2	
gp_i	[30,90]		w_g	0.6	
Loc_k	[0,500]	km	w_e	0.8	
N_k	[1100]		w_b	0.1	
WT_k	[10,30]	kg	w_a	0.3	
rel_i	[0.8,1)				

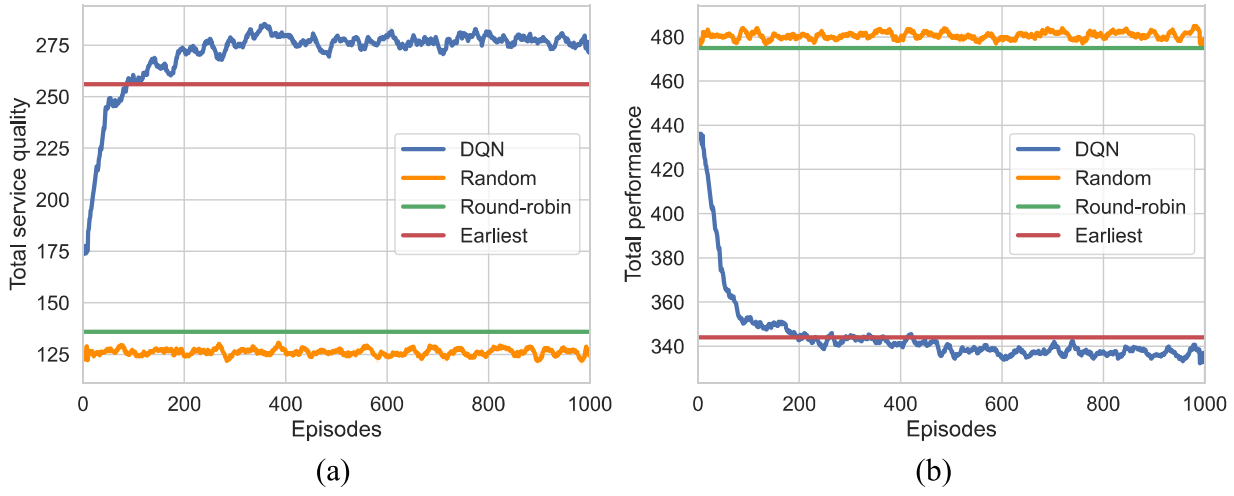

Fig. 2. Comparison of different algorithms.

Table 3
Information of the first ten robot services (due to space limitation, only the information of the first ten robot services is provided).

	t_i	f_i	p_i	lp_i	bp_i	ap_i	gp_i
R_1	t_1(7.891)	f_1(1)	p_1(87.364)	lp_1(110,084.372)	bp_1(43)	ap_1(53)	gp_1(82)
R_2	t_2(8.922)	f_2(3)	p_2(38.809)	lp_2(122,181.179)	bp_2(75)	ap_2(70)	gp_2(35)
R_3	t_3(9.109)	f_3(0)	p_3(87.47)	lp_3(91,377.357)	bp_3(72)	ap_3(53)	gp_3(43)
R_4	t_4(8.234)	f_4(1)	p_4(50.348)	lp_4(129,306.553)	bp_4(53)	ap_4(83)	gp_4(30)
R_5	t_5(5.898)	f_5(2)	p_5(33.559)	lp_5(100,764.888)	bp_5(55)	ap_5(42)	gp_5(73)
R_6	t_6(8.631)	f_6(8)	p_6(79.154)	lp_6(92,186.476)	bp_6(51)	ap_6(74)	gp_6(88)
R_7	t_7(6.424)	f_7(0)	p_7(78.204)	lp_7(88,230.576)	bp_7(71)	ap_7(49)	gp_7(36)
R_8	t_8(9.207)	f_8(0)	p_8(57.799)	lp_8(100,260.910)	bp_8(76)	ap_8(74)	gp_8(36)
R_9	t_9(7.851)	f_9(6)	p_9(76.341)	lp_9(95,467.186)	bp_9(55)	ap_9(55)	gp_9(41)
R_{10}	t_{10}(5.426)	f_{10}(2)	p_{10}(74.929)	lp_{10}(111,210.631)	bp_{10}(75)	ap_{10}(75)	gp_{10}(75)

Table 4
Information of the first ten robot services (continued).

	rel_i	ep_i	qp_i	L_j^l	loc_i
R_1	rel_1(0.805)	ep_1(0.972)	qp_1(0.028)	L_3^l(3)	loc_1(446.430,165.990)
R_2	rel_2(0.825)	ep_2(0.903)	qp_2(0.097)	L_4^l(4)	loc_2(410.614,20.848)
R_3	rel_3(0.962)	ep_3(0.997)	qp_3(0.003)	L_5^l(5)	loc_3(53.828,297.526)
R_4	rel_4(0.976)	ep_4(0.869)	qp_4(0.131)	L_4^l(4)	loc_4(264.909,209.404)
R_5	rel_5(0.944)	ep_5(0.985)	qp_5(0.015)	L_3^l(3)	loc_5(167.704,311.260)
R_6	rel_6(0.844)	ep_6(0.875)	qp_6(0.125)	L_1^l(1)	loc_6(219.071,367.941)
R_7	rel_7(0.843)	ep_7(0.896)	qp_7(0.104)	L_3^l(3)	loc_7(259.022,289.438)
R_8	rel_8(0.898)	ep_8(0.955)	qp_8(0.045)	L_4^l(4)	loc_8(322.689,495.119)
R_9	rel_9(0.877)	ep_9(0.812)	qp_9(0.188)	L_4^l(4)	loc_9(409.936,206.609)
R_{10}	rel_{10}(0.833)	ep_{10}(0.951)	qp_{10}(0.049)	L_4^l(4)	loc_{10}(438.136,411.888)

Table 5
Information of logistics services.

	p_j^l	v_j^l	sl_j^l	λ_j^l	rel_j^l	loc_j^l
L_1^l	p_1^l(0.093)	v_1^l(96.125)	sl_1^l(0.831)	λ_1^l(0.614)	rel_1^l(0.824)	loc_1^l(208.789,371.245)
L_2^l	p_2^l(0.145)	v_2^l(85.145)	sl_2^l(0.476)	λ_2^l(0.787)	rel_2^l(0.908)	loc_2^l(346.476,5.835)
L_3^l	p_3^l(0.183)	v_3^l(92.745)	sl_3^l(0.660)	λ_3^l(0.837)	rel_3^l(0.895)	loc_3^l(192.083,438.926)
L_4^l	p_4^l(0.192)	v_4^l(81.589)	sl_4^l(0.429)	λ_4^l(0.316)	rel_4^l(0.839)	loc_4^l(373.120,91.493)
L_5^l	p_5^l(0.110)	v_5^l(81.895)	sl_5^l(0.596)	λ_5^l(0.134)	rel_5^l(0.895)	loc_5^l(249.377,448.399)

Table 6
Partial information of the first ten tasks.

	O_k	N_k	Ft_k	WT_k	Loc_k
C_1	O_1	N_1(27)	Ft_1(9)	WT_1(13.826)	Loc_1(149.945,188.255)
C_2	O_2	N_2(79)	Ft_2(0)	WT_2(27.653)	Loc_2(367.714,172.763)
C_3	O_3	N_3(77)	Ft_3(4)	WT_3(24.399)	Loc_3(365.047,338.747)
C_4	O_4	N_4(39)	Ft_4(6)	WT_4(26.237)	Loc_4(44.746,360.152)
C_5	O_5	N_5(2)	Ft_5(8)	WT_5(25.367)	Loc_5(28.809,122.265)
C_6	O_6	N_6(79)	Ft_6(1)	WT_6(27.600)	Loc_6(427.085,497.960)
C_7	O_7	N_7(66)	Ft_7(3)	WT_7(28.470)	Loc_7(429.144,377.372)
C_8	O_8	N_8(2)	Ft_8(5)	WT_8(11.160)	Loc_8(135.157,54.906)
C_9	O_9	N_9(62)	Ft_9(6)	WT_9(24.487)	Loc_9(143.117,76.463)
C_{10}	O_{10}	N_{10}(72)	Ft_{10}(6)	WT_{10}(27.575)	Loc_{10}(314.886,52.130)

Table 7
Hyper-parameters for the training of DQN and DDQN.

Parameters	Value	Description
N	3000	The capacity of experience replay buffer
b	30	Number of batches per training step
l	0.01	Initial learning rate
γ	0.9	Discounted factor
C	100	Target network update frequency
ε_{ini}	0.002	Probability of initial exploration
ε_{end}	0.9	Probability of final exploration

$$Et_{i,k} = N_k \times t_i \tag{10}$$

The waiting time $Wt_{i,k}$ for R_i to complete O_k depends on $t_{i,k}$ and N_k, which can be calculated as follows:

$$Wt_{i,k} = t_{i,k} + 2 \times t_{i,k} + \cdots + (N_k - 1) \times t_{i,k} = \sum_{n=1}^{N_k - 1} n \times t_{i,k} \tag{11}$$

There are two stages for logistics. The first stage is from the location of raw materials or part that need to be processed to that of the enterprise offering robot services, and can be calculated below:

$$Lt_{i,l_i}^l = d\left(loc_{l_i}^l, loc_i\right) \Big/ v_{l_i}^l \tag{12}$$

where $d(loc_{l_i}^l, loc_i)$ is the geographical distance between $loc_{l_i}^l$ and loc_i. The second stage is from the location of the enterprise that provides the robot service to that of the consumer, and can be calculated as:

$$Lt_{i,k}^l = d(loc_i, Loc_k) \Big/ v_{l_i}^l \tag{13}$$

The total logistics time $Lt_{i,l_i,k}^l$ can be computed below.

$$Lt_{i,l_i,k}^l = max\left(Et_{k,i}, Lt_{i,l_i}^l\right) + Lt_{i,k}^l \tag{14}$$

The total completion time for R_i to complete O_k can be computed according to the following formula.

$$Tt_{i,k} = Et_{k,i} + Wt_{k,i} + Lt_{i,l_i,k}^l \tag{15}$$

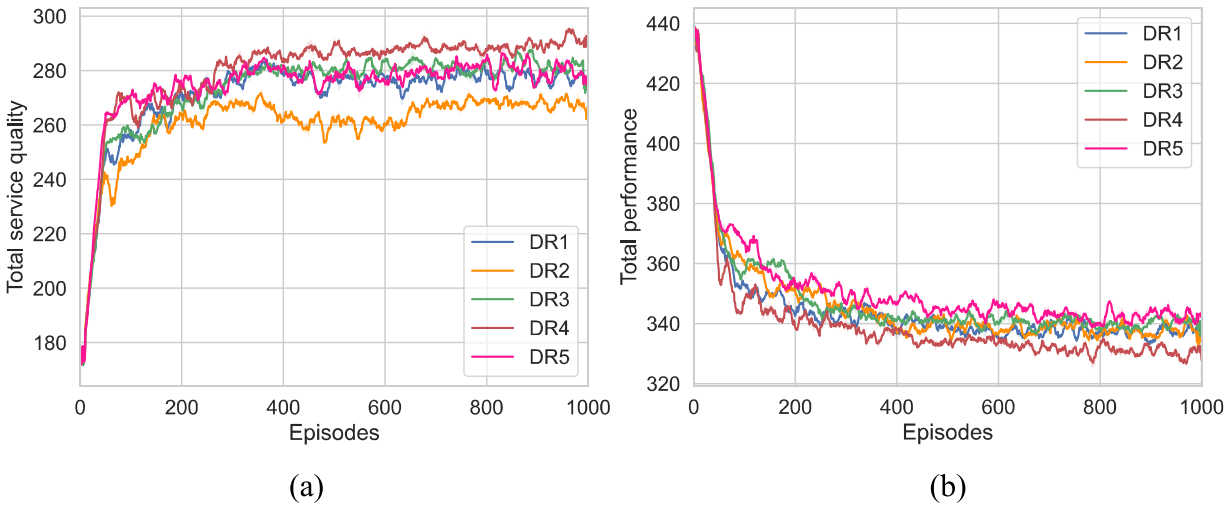

Fig. 3. Effects of different combinations of weight coefficients of relevant indicators on total service quality and total performance.

Table 8
Indicators of different ratios.

	w_{lp}	w_{sf}	w_{qg}	w_{rel}	w_{rel^l}	w_{Tp}	w_{Tt}	w_{Sp}
DR1	0.2	0.2	0.2	0.2	0.2	0.33	0.33	0.33
DR2	0.1	0.1	0.1	0.6	0.1	0.33	0.33	0.33
DR3	0.1	0.1	0.1	0.1	0.6	0.33	0.33	0.33
DR4	0.2	0.2	0.2	0.2	0.2	0.25	0.25	0.5
DR5	0.2	0.2	0.2	0.2	0.2	0.25	0.5	0.25

(2) Total price

The total price includes that of robot services and logistics services. Robot service price $Rp_{i,k}$ for R_i to complete O_k depends on p_i and N_k, and can be computed by Eq. (16).

$$Rp_{i,k} = N_k \times p_i \tag{16}$$

As with calculation of logistics time, there are also two part for logistics price $Lp^l_{i,l_i,k}$, which can be calculated by Eqs. (17) and (18).

$$Lp^l_{i,l_i} = p_i \times d\left(loc^l_{l_i}, loc_i\right) \tag{17}$$

$$Lp^l_{i,k} = p_i \times d(loc_i, Loc_k) \times WT_k \times N_k \tag{18}$$

The total logistics price is:

$$Lp^l_{i,l_i,k} = Lp^l_{i,l_i} + Lp^l_{i,k} \tag{19}$$

Therefore, the total price for R_i to complete O_k can be computed by Eq. (20).

$$Tp_{i,l_i,k} = Rp_{i,k} + Lp^l_{i,l_i,k} \tag{20}$$

(3) Risk probability

Inspired by [48], the risk probability $Sp^l_{i,l_i,k}$ to quantitatively characterize the performance of $L^l_{l_i}$ for O_k can be calculated by Eq. (21).

$$Sp^l_{i,l_i,k} = 1 - e^{-\lambda^l_{l_i}\left(1-sl^l_{i,l_i,k}\right)} \tag{21}$$

where $sl^l_{i,l_i,k}$ is security level of $L^l_{l_i}$, and $\lambda^l_{l_i}$ is the security coefficient of $L^l_{l_i}$.

3.3.3. Scheduling objective

The scheduling objective is to realize the maximization of total service quality and minimization of total service performance.

(1) Total service quality

Maximizing the total service quality can be defined by Eq. (22).

$$MaxTQ = \sum_{i=1}^{I} n_i \times \left(RQ_i + LQ^l_{l_i}\right) = \sum_{i=1}^{I} n_i \times \left(lp_i + sf_i + qg_i + rel_i + rel^l_{l_i}\right) \tag{22}$$

where n_i are number of task executed in R_i, TQ is a multi-objective function of the variables of RQ and LQ.

(2) Total performance

Minimizing the total service performance can be defined by Eq. (23).

$$\begin{aligned} MinTP &= \sum_{i=1}^{I}\left(RP_i + LP^l_{l_i}\right) = \sum_{i=1}^{I}\left(Tp_i + Tt_i + Tp^l_{l_i} + Ms^l_{l_i} + Sp^l_{l_i}\right) \\ &\sum_{i=1}^{I}\sum_{k=0}^{n_i}\left(Rp_{i,k} + Et_{i,k} + Wt_{i,k} + Lp^l_{i,l_i,k} + Lt^l_{i,l_i,k} + Sp^l_{i,l_i,k}\right) \\ &= \sum_{i=1}^{I}\sum_{k=0}^{n_i}\left(Tp_{i,l_i,k} + Ms_{i,l_i k} + Sp^l_{i,l_i,k}\right) \end{aligned} \tag{23}$$

where TP is a multi-objective function of the variables of RP and LP, n_i and l_i have the same meaning as before.

In order to avoid the non-standardization error that brings about the unrealistic results, the values of different indexes are normalized between 0 and 1 using the following methods [17]:

$$u' = \frac{u - u_{min}}{u_{max} - u_{min}} \tag{24}$$

$$u' = \frac{u_{max} - u}{u_{max} - u_{min}} \tag{25}$$

where Eq. (24) is adopted for positive attributes (the larger the better) like RQ and LQ, and Eq. (25) is adopted for negative attributes (the lower the better) like RP and LP. The values u_{min} and u_{max} are, respectively, the minimum and maximum values of the corresponding indexes.

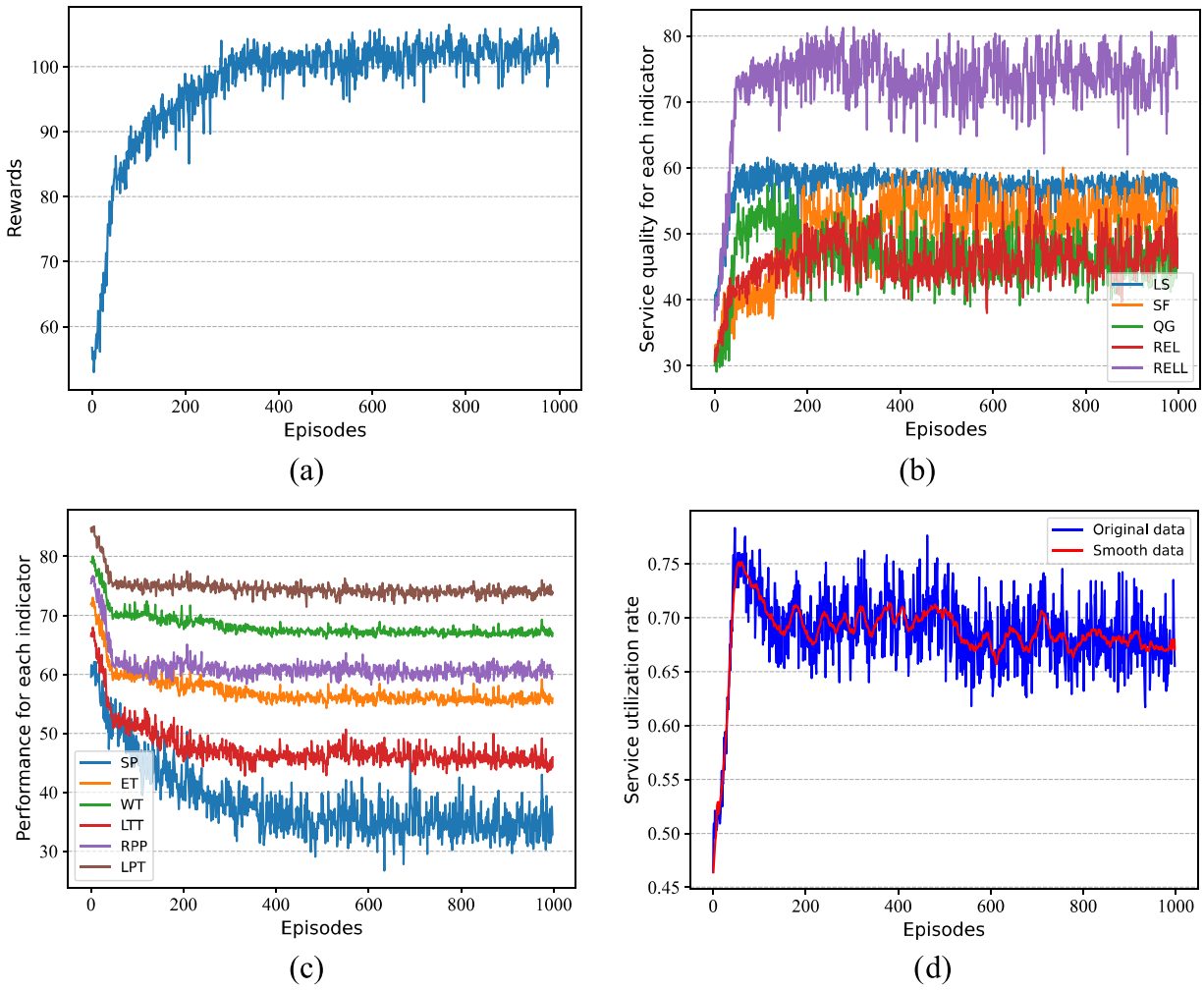

Fig. 4. Results of DQN-based scheduling algorithm.

Table 9
Intercept and regression coefficients of different indicators.

	$\overline{\omega_{sf}}$	$\overline{\omega_{qg}}$	$\overline{\omega_{rel}}$	$\overline{\omega_{rel'}}$	e_{TQ}
1.00267	0.99384	0.99664	0.98062	1.00869	2.52907

Table 10
Intercept and regression coefficients of different indicator of total performance.

	$\overline{\omega_{Et}}$	$\overline{\omega_{Wt}}$	$\overline{\omega_{Lp^i}}$	$\overline{\omega_{Lt^i}}$	$\overline{\omega_{Sp^i}}$	e_{TP}
0.97035	1.22560	0.68369	0.73595	0.98115	1.04139	31.71739

4. DDQN-based scheduling algorithm

4.1. DRL-based scheduling in cloud manufacturing

Fig. 1 presents a schematic diagram of DRL-based scheduling in cloud manufacturing. The overall scheduling process is as follows. First of all, the cloud manufacturing environment produces information as state inputs to DNN. The scheduling agent interacts with the cloud manufacturing environment and takes actions to the environment via DNN according to the information. Finally, the cloud manufacturing environment transitions into another state and yields a reward. This training loop (i.e., state-action-reward) repeats until the terminal state or the maximum time step is reached.

Fig. 2

4.2. State space, action space, and reward function

The process in Fig. 1 can be modeled as a Markov Decision Process (MDP), where at each iteration, the cloud manufacturing environment returns a reward to the scheduling agent. The scheduling agent is trained over many iterations to learn the optimal policy. The DNN is updated each time an iteration finishes.

(1) State space and state transition

States in the scheduling system include that of tasks and services. The state of O_k includes its arrival time At_k and functional type Ft_k, and services' state is characterized by the waiting time of all candidate services for T_k. Therefore, the state space can be described as follows.

$$S = \left\{At_k, Ft_k, Wt_{k,1}, Wt_{k,2}, \cdots, Wt_{k,n_k}\right\} \tag{26}$$

where Wt_{k,n_k} is the waiting time of O_k for the $n_k th$ candidate service, and n_k is the total number of candidate services for O_k.

Each time a task is completed, the current state transitions to the next one. The final state is reached when scheduling of all tasks completes. The above process from the first state to the final one is called an episode. The training process iterates episode by episode until it terminates.

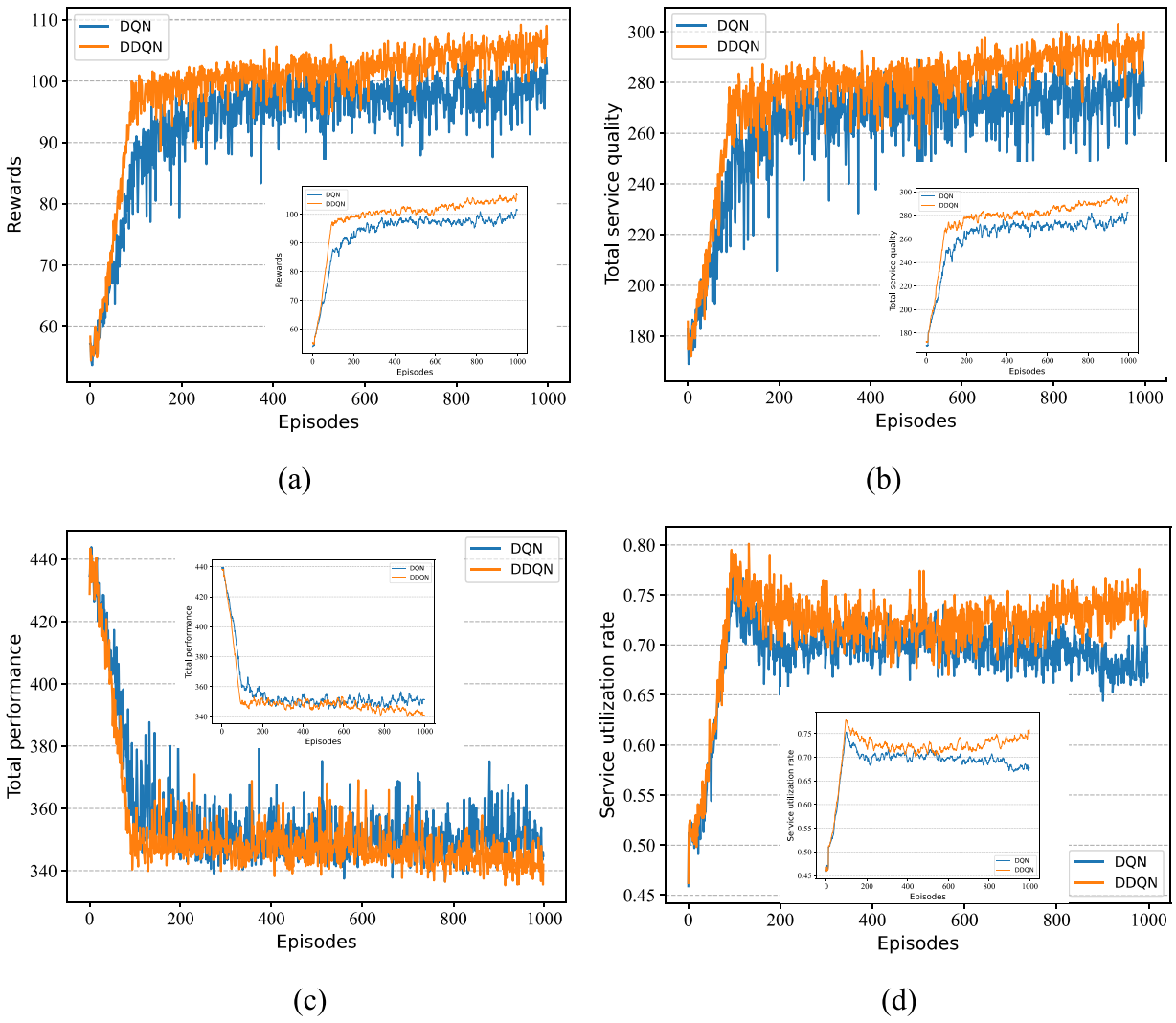

Fig. 5. Comparison of performance of the DQN- and DDQN-based scheduling approaches.

(2) Action space and action selection

An action of O_k is represented by a candidate robot service, and therefore the action space is represented by the candidate service set of O_k.

$$A = \left\{R_{k,1}, R_{k,2}, \cdots, R_{k,n_k}\right\} \tag{27}$$

where R_{k,n_k} is the $n_k th$ candidate service of O_k.

The training process repeats from one episode to another until all episodes terminate. In each episode, all of the tasks have the chance to be scheduled. Action selection relies on the current state and the estimated Q-value. The ε-greedy policy is adopted for the training.

(3) Reward function

The reward is designed according to Eq. (28):

$$Reward = \begin{cases} \dfrac{w_{lp}nlp_{i,k} + w_{sf}nsf_{i,k} + w_{qg}nqg_{i,k} + w_{rel}nrel_{i,k} + w_{rel'}nrel^{l}_{i,l_i,k}}{w_{Tp}nTp_{i,l_i,k} + w_{Tt}nTt_{i,l_i,k} + w_{Sp}nSp^{l}_{i,l_i,k}}, & if\ k < K \\ \dfrac{w_{lp}nlp_{i,k} + w_{sf}nsf_{i,k} + w_{qg}nqg_{i,k} + w_{rel}nrel_{i,k} + w_{rel'}nrel^{l}_{i,l_i,k}}{w_{Tp}nTp_{i,l_i,k} + w_{Tt}nTt_{i,l_i,k} + w_{Sp}nSp^{l}_{i,l_i,k}} + \dfrac{I_k}{I}, & if\ k = K \end{cases} \tag{28}$$

where $nlp_{i,k}$, $nsf_{i,k}$, $nqg_{i,k}$, and $nrel_{i,k}$ are the normalized life span, normalized specifications, normalized quality grade, and normalized reliability for the kth task executed on robot service R_i, respectively, $nrel^{l}_{i,l_i,k}$ is logistics service reliability for $L^{l}_{l_i}$ to transport O_k, $nTp_{i,l_i,k}$, $nTt_{i,l_i,k}$, and $nSp^{l}_{i,l_i,k}$ are, respectively, the normalized total price, the normalized total completion time, and the normalized risk probability for R_i to complete O_k. By means of the method of analytic hierarchy process (AHP), we can obtain the weights of all indicators of service quality (i.e., w_{lp}, w_{sf}, w_{qg}, w_{rel}, and $w_{rel'}$). Similarly, the weights of indicators of total performance including w_{Tp}, w_{Tt}, and w_{Sp} can be obtained by means of AHP. I_k is the number of robot services occupied when scheduling of all tasks are completed. Note that the second reward function in Eq. (28) is used when the all tasks is completed (i.e. $k = K$).

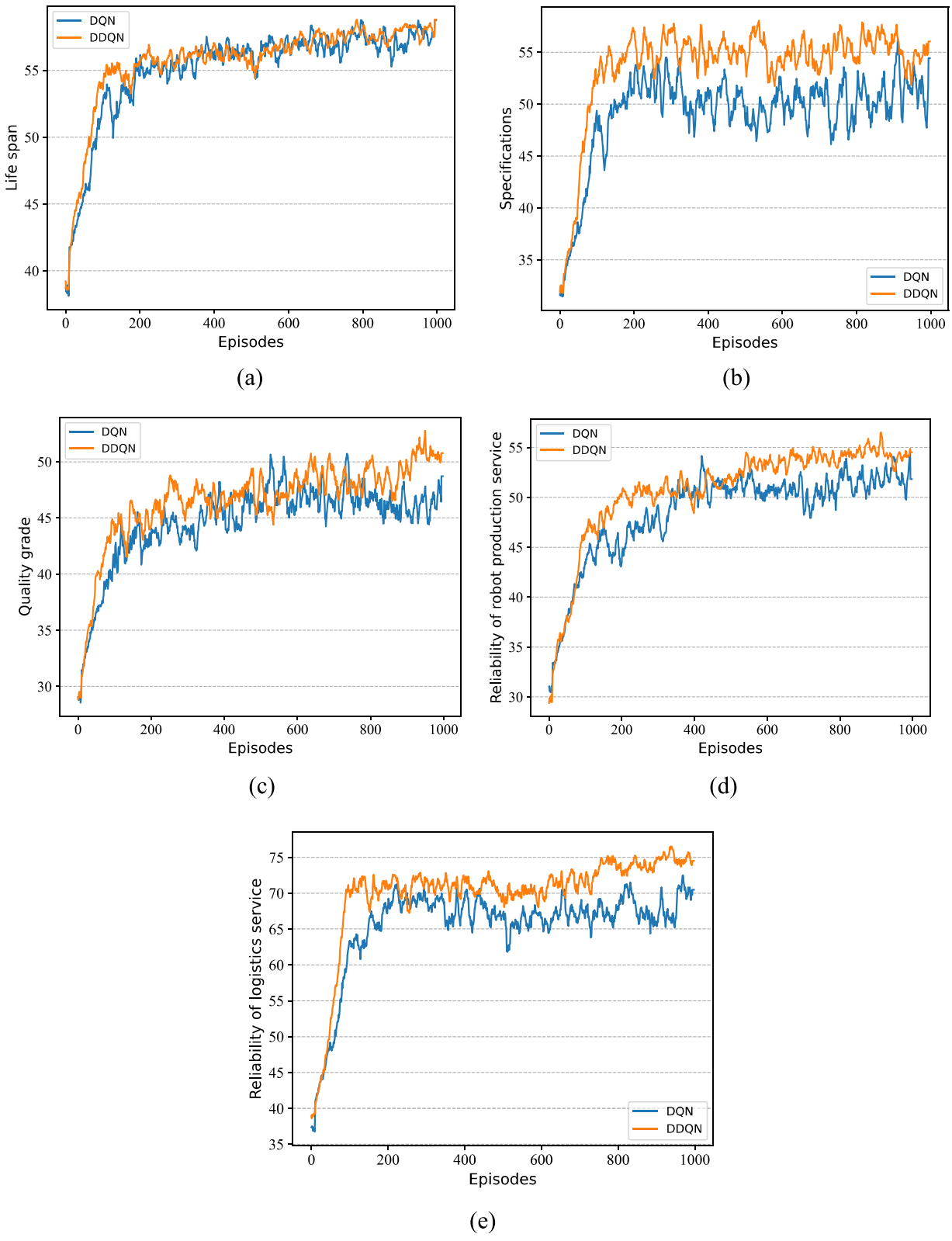

Fig. 6. Comparison of values of different indicators in the total service quality for DQN and DDQN.

4.3. DDQN and DDQN-based scheduling algorithm

For some states, actions are independent of the expected values. There is no need to learn how these states affect actions. That is, these actions have little impact on the expected values. Therefore, decoupling the state value and Q value of irrelevant action can lead to more robust learning effect. Different from DQN, DDQN modifies the network architecture and the Q value is divided into two parts: state value and action advantage, which can be described by Eq. (29).

$$Q^{\pi}(s,a) = V^{\pi}(s) + A^{\pi}(s,a) \tag{29}$$

DDQN not only evaluates the value $Q^{\pi}(s,a)$ of an action in a certain

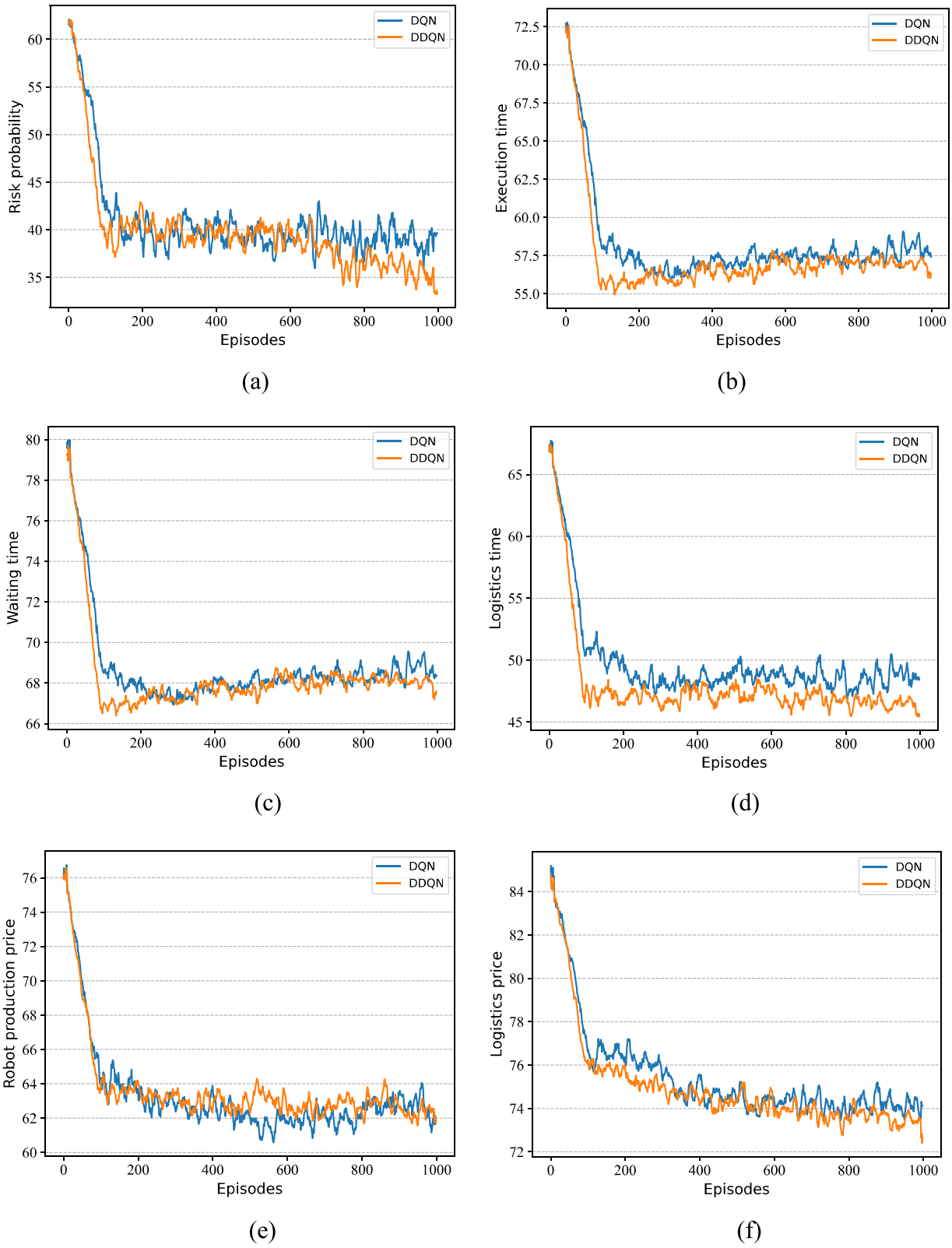

Fig. 7. Comparison of values of different indicators in the total service performance for DQN and DDQN.

state, but also evaluates the value function $V^{\pi}(s)$ of the state and the relative value function $A^{\pi}(s,a)$ of each action in this state. DDQN determines the Q value for each action and can be described by Eq. (30).

$$Q^{\pi}(s,a;\theta,\alpha,\beta) = V^{\pi}(s;\theta,\alpha) + \left(A^{\pi}(s,a;\theta,\beta) - \max_{a'} A^{\pi}(s,a';\theta,\beta)\right) \tag{30}$$

where α and β are parameters for two full connection layers, θ is the parameter of the convolution layer, *max* function ensures that the Q value uniquely corresponds to the corresponding state value and action advantage; otherwise, the training ignores the state values and only makes advantage function converge to Q value. In order to obtain more

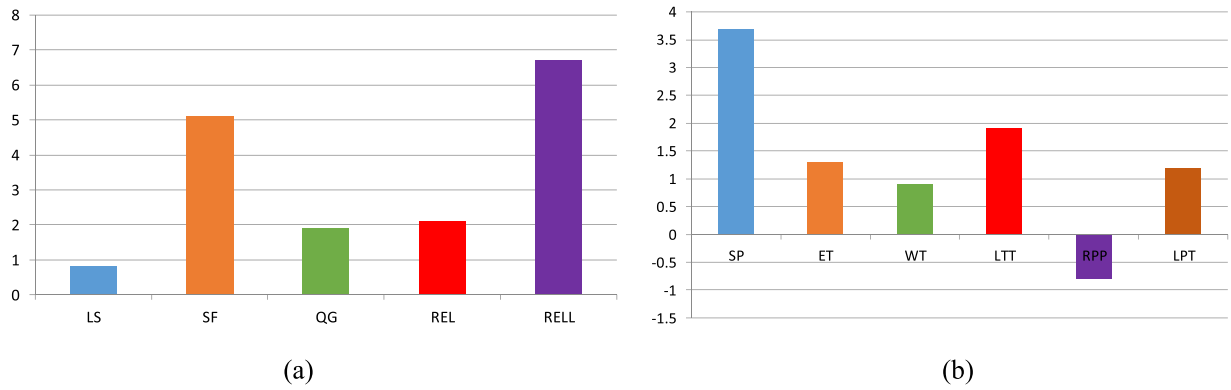

Fig. 8. Comparison of average difference values of different indicators.

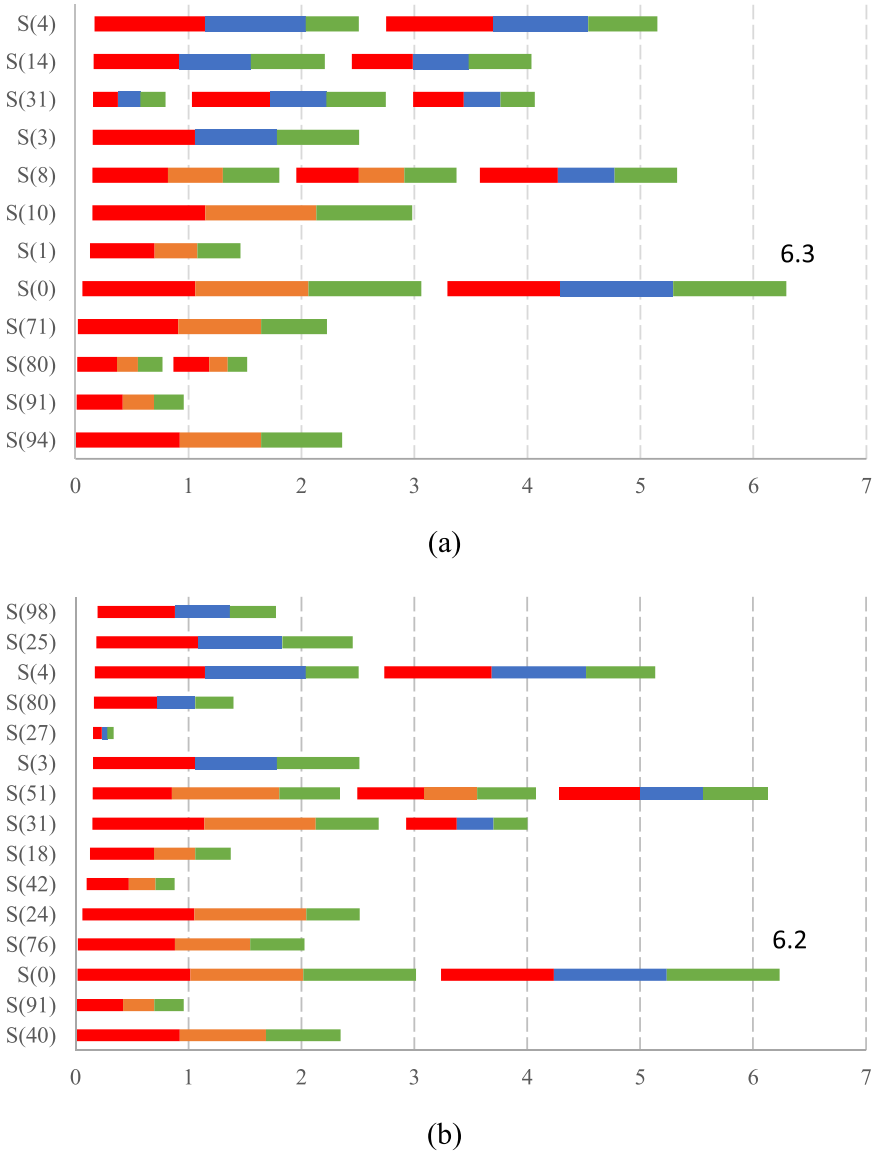

Fig. 9. Scheduling diagrams of the first 20 tasks for (a) DQN- and (b) DDQN-based scheduling approach.

stable convergence results, the average method replaces the maximum method [34], and can be defined is described by Eq. (31).

$$Q^{\pi}(s,a,\theta,\alpha,\beta) = V^{\pi}(s;\theta,\alpha) + \left(A^{\pi}(s,a;\theta,\beta) - \frac{1}{|A|}\sum_{a'} A^{\pi}(s,a';\theta,\beta)\right) \quad (31)$$

where $|A|$ is the number of a discrete action set. As described above, the advantage function is close to the mean and training can obtain stable convergence results.

The pseudo-code of the DDQN-based scheduling algorithm of decentralized robot services is shown in Table 1.

5. Case study

This section presents a case study of scheduling of decentralized cloud manufacturing that involves 100 tasks, 100 robot services, and 5 logistics services. The number of tasks in each order of a consumer follows a uniform distribution between 1 and 100. Model variables and parameters are shown in Tables 2-7.

Three benchmark scheduling approaches, i.e., random scheduling, round-robin scheduling, earliest scheduling, and DQN are also considered for comparison. For random scheduling, a service for a task is selected randomly among the matched services. For the round-robin scheduling, robot services are dispatched to different tasks fairly. The earliest scheduling means that the earliest available service is always selected for a task during scheduling.

The experimental results are conducted on a computer with configuration of Intel Core i7 2.90 GHz and 32.0 GB of memory, and a 64-bit operating system based on a x64 processor module. All of the simulation programs are written with Python 3.7.4 version.

The dueling network proposed consists of a target network and an evaluate network. Each DNN has an input layer and two fully connected hidden layer, and each layer has 20 neurons. The last layer is divided into two parts: one evaluating the value of an action in a certain state and the other layer evaluates the value function, and each layer has 20 neurons as well. The remaining parameters are shown in Table 7. Each result is obtained by averaging 10 steps over 10,000 episodes.

5.1. Simulation results and analysis

5.1.1. Model training

(1) Performance of different algorithms

Fig. 3(a) and (b) illustrate the variations of the total quality (i.e. sum of RQ and LQ) and the total performance (i.e. sum of RP and LP) with increasing episodes, respectively. In each episode, scheduling is conducted across all tasks. The total service quality and the total performance are defined as the sum of the service quality and performance of all tasks in each episode, respectively. The scheduling objective is to maximize the total service quality and minimize the total performance. Robot services are fairly selected by the round-robin approach and the earliest available services are deployed by earliest approach in each episode, therefore the results of the two approaches are constant. The curves obtained with the random approach fluctuate with under small amplitude. For the DQN algorithm, the curve of the total quality gradually increases and finally reaches a dynamic stable state, while that of the total performance gradually decreases and finally stabilizes. The results indicate that the DQN-based scheduling approach performs the best among all of the algorithms.

(2) Weight coefficients of relevant indicators

Fig. 3 shows the effects of different combinations of weight coefficients of relevant indicators (Table 8) on the total service quality and the total performance. The results indicate that the combinations of DR2 and DR3 lead to a drastic change in the total service quality, and the total service quality of DR2 is approximately 20 less than that of DR3. However, the changes of DR4 and DR5 in the total service quality are relatively less prominent. Similarly, the total performance of DR4 is about 20 less than that of DR5. DR2 and DR3 make total performance generate a slight change. For the sake of simplicity but without loss of generality, the following comparison is based on DR1.

(3) DQN-based scheduling algorithm

The section presents simulation results of the DQN scheduling algorithms. Fig. 4(a) shows the rewards versus episodes, in which each data point is the sum of the rewards of all tasks in an episode. In Fig. 4 (b), the service quality of different indicators, including life span (LS), specifications (SF), quality grade (QR), reliability of robot service (REL), and reliability of logistics services (RELL), are shown. The results indicate that overall the values of service quality of all of the indicators increase monotonously, and in particular, there is a huge increase for RELL. For other indicators (i.e. LS, SF, QR, and REL), the increase is moderate. Fig. 4(c) shows the performance of different indicators (i.e. risk probability (SP), execution time (ET), waiting time (WT), logistics time (LTT), robot production price (RPP), and logistics price (LPT)). The results indicate that overall the performance of the different indicators (i.e. SP, ET, WT, LTT, RPP, and LPT) have a downward trend. There is a huge decrease in SP. In addition, other indicators (i.e. ET, WT, LTT, RPP, and LPT) contain a moderate decline. Fig. 4(d) shows the service utilization rate, which is a little low because some preponderant services cannot be obtained. The DDQN-based scheduling algorithm can solve this issue.

(4) Influencing degrees of different indicators

This section uses a multiple linear regression method to analyze the influencing degrees of different indicators. For the total service quality, a multiple linear regression formula is designed, the optimal intercept and regression coefficient of each indicator are obtained, and the corresponding formula is as follows.

$$TQ = \sum_{i=1}^{I} n_i \times \left(\overline{\omega_{lp}} lp_i + \overline{\omega_{sf}} sf_i + \overline{\omega_{qg}} qg_i + \overline{\omega_{rel}} rel_i + \overline{\omega_{rel^l}} rel^l_{l_i}\right) + e_{TQ} \quad (32)$$

where $\overline{\omega_{lp}}$, $\overline{\omega_{sf}}$, ω_{qr}, $\overline{\omega_{rel}}$, $\overline{\omega_{rel^l}}$ are, respectively, the weight coefficients of the corresponding indicators, and e_{TQ} is the intercept. The corresponding values are shown in Table 9.

Table 9 indicates that the maximum weight coefficient is $\overline{\omega_{rel^l}}$, revealing that the reliability of logistics services is of the greatest influence on the total service quality.

For the total performance, the following multiple linear regression formula is designed.

$$TP = \sum_{i=1}^{I}\sum_{k=0}^{n_i}\left(\overline{\omega_{Rp}} Rp_{i,k} + \overline{\omega_{Et}} Et_{i,k} + \overline{\omega_{Wt}} Wt_{i,k} + \overline{\omega_{Lp^l}} Lp^l_{i,l_i,k} + \overline{\omega_{Lt^l}} Lt^l_{i,l_i,k} + \overline{\omega_{Sp^l}} Sp^l_{i,l_i,k}\right) + e_{TP} \quad (33)$$

where $\overline{\omega_{Rp}}$, $\overline{\omega_{Et}}$, $\overline{\omega_{Et}}$, $\overline{\omega_{Lp^l}}$, $\overline{\omega_{Lt^l}}$, $\overline{\omega_{Sp^l}}$ are, respectively, the weight coefficients of the corresponding indicators, and e_{TP} is the intercept. The corresponding values are as follows.

Table 10 shows that the maximum weight coefficient is $\overline{\omega_{Et}}$, revealing that the execution time has the greatest influence on the total performance.

5.1.2. DDQN-based task scheduling

(1) Global objective analysis

In order to obtain the preponderant action, a DDQN-based scheduling algorithm is proposed. Performance of the DQN-based scheduling algorithm is compared with that of the DDQN-based scheduling algorithm, including rewards, total service quality, total performance, and service utilization rate, as shown in Fig. 5(a)-(d), where each data point is an average of 10 episodes over the 10,000 episodes and the embedded figures adopt a moving average method that aims to filter out high-frequency perturbations in each episode and retain useful low-frequency data.

Comparisons of results obtained with DQN- and DDQN-based scheduling approaches indicate that: (1) the DDQN-based scheduling approach leads to a higher reward and stabilizes more quickly than the DQN-based approach, (2) the latter is also better than the former in terms of the total service quality and service utilization rate, and (3) the latter also outperforms the former in term of total performance.

(2) Analysis of different indicators

The subsection presents a detailed comparison of values of different indicators in the total service quality for DQN and DDQN, as shown in Fig. 6(a)-(e). It can be observed that except for LS, the DDQN-based scheduling algorithm performs apparently better than the DQN-based scheduling algorithm for all of the other indicators (i.e. SF, QG, REL, and RELL), reflected by the fact that the orange curve (DDQN) for each indicator is higher than the blue one (DQN).

Fig. 7(a)-(f) show the comparison of results of different indicators (including SP, ET, WT, LTT, RPP, and LPT) in the total service performance for DQN and DDQN. It can be observed that except for WT and RPP, for all other indicators (i.e. SP, ET, LTT, and LPT), the DDQN-based scheduling approach performs relatively better that the DQN-based scheduling approach.

In order to clearly compare each indicator of the two algorithms, Fig. 8 shows the comparison of average difference values of the different indicators for the two scheduling algorithms in last 200 episodes. The result indicates that although DDQN is a little bit lower than DQN in the robot service price, DDQN outperforms DQN in the majority of indicators. Note that the multi-objective interference generates a negative number, which leads to the fact that DDQN is slightly lower than DQN in this indicator.

Fig. 9 presents the scheduling diagrams of the two scheduling algorithms. For the sake of simplicity but without loss of generality, only the first 20 tasks are considered. The brown squares represent the execution time of the first 10 tasks, the blue squares represent the execution time of the subsequent 10 tasks. The red and green squares represent their waiting time and logistics time, respectively. The result indicates that DDQN scheduling algorithm outperforms that of DQN.

The current paper could provide some managerial implications for managers and decision-makers, which lie in the following aspects: (1) the proposed DRL-based scheduling algorithms could provide some insights for the platform operator to manage the platform so as to satisfy consumers' robot services requirements on demand. (2) The results obtained with different combinations of weight coefficients could provide some reference for managers and decision-makers to determine the optimal combinations of weight coefficients. (3) The insights result from analysis of the influencing degrees of different indicators enable managers and decision-makers to identify the key indicators for making optimal decisions.

6. Conclusions

In this paper, focusing on robot services in cloud manufacturing, a novel scheduling model based on DRL, including both DQN and DDQN, was proposed. Quality and performance of both robot services and logistics services were considered. The objective was to realize maximization of service quality and minimization of service performance. The performance of the DDQN-based scheduling approach was compared with that DQN and other three benchmark scheduling approach. Comparison indicated that DDQN-based scheduling approach outperforms all of the other approach with respect to each index. A multiple linear regression method was used to analyze the influencing degrees of different indicators, and the results indicated that the reliability of logistics services and execution times are the most influential indicators. The current paper could provide some managerial implications for managers and decision-makers, which lie in the following aspects: (1) the proposed DRL-based scheduling algorithms could provide some insights for the platform operator to manage the platform so as to satisfy consumers' robot services requirements on demand. (2) The results obtained with different combinations of weight coefficients could provide some reference for managers and decision-makers to determine the optimal combinations of weight coefficients. (3) The insights result from analysis of the influencing degrees of different indicators enable managers and decision-makers to identify the key indicators for making optimal decisions.

In the future, efforts could be made towards the following related issues. First, dynamic scheduling in cloud manufacturing where dynamic events such as robot breakdowns and requirements changes occur need to be investigated with DRL. Second, how to efficiently explore the advantages of DRL (e.g. selection of DNN, design of reward function, and optimization of loss function) to acquire optimal scheduling policies is also an important problem. Third, with the increasing demand for cloud-edge collaboration in cloud manufacturing, exploring DRL-based scheduling solutions in the context of cloud-edge collaboration is worthy of study.

Author statement

Dear Editor-in-Chief Prof. Lihui Wang and Reviewers,

Thank you very much for handling and reviewing our paper, and providing us with helpful comments that help us improve this paper significantly.

We have revised the manuscript accordingly. Please see the blow for detailed responses and revisions.

The parts modified are highlighted in BLUE in the revised manuscript.

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.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61973243 and 61873014, and Technological Innovation 2025 Major Project by Ningbo Science and Technology Bureau under Grant No. 2019B10081.

References

[1] E.G. Talbi, Metaheuristics: From Design to Implementation, John Wiley and Sons, 2009.
[2] T. Gonzalez, Handbook of Approximation Algorithms and Metaheuristics, Chapman and Hall, London, 2007.
[3] W. Li, C. Zhu, L.T. Yang, et al., Subtask scheduling for distributed robots in cloud manufacturing, IEEE Syst. J. 11 (2) (2017) 941–950.
[4] H. Akbaripour, M. Houshmand, T.V. Woensel, et al., Cloud manufacturing service selection optimization and scheduling with transportation considerations: mixed-integer programming models, Int. J. Adv. Manuf. Technol. 95 (1) (2018) 43–70.

[5] V. Mnih, A.P. Badia, M. Mirza, et al., Asynchronous methods for deep reinforcement learning, Int. Conf. Mach. Learn. (2016) 1928, 1037.
[6] V. Mnih, K. Kavukcuoglu, D. Silver, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[7] H. Liang, X. Wen, Y. Liu, et al., Logistics-Involved QoS-Aware Service Composition in Cloud Manufacturing With Deep Reinforcement Learning, Robotics and Computer-Integrated Manufacturing, 2021, p. 67.
[8] A. Halty, R. Sánchez, V. Vázquez, et al., Rossit, Scheduling in cloud manufacturing systems: recent systematic literature review, Math. Biosci. Eng. 17 (6) (2020) 7378–7397.
[9] Y. Liu, X. Xu, L. Zhang, et al., Workload-based multi-task scheduling in cloud manufacturing, Robot. Comput. Integr. Manuf. 45 (2017) 3–20.
[10] Y. Liu, X. Zhang, L. Zhang, et al., A multi-agent architecture for scheduling in platform-based smart manufacturing systems, Front. Inf. Technol. Electr. Eng. 20 (11) (2019) 1465–1492.
[11] Y. Liu, L. Wang, Y. Wang, et al., Multi-agent-based scheduling in cloud manufacturing with dynamic task arrivals, Procedia CIRP 72 (2018) 953–960.
[12] L. Zhou, L. Zhang, A dynamic task scheduling method based on simulation in cloud manufacturing, in: Asian Simulation Conference SCS Autumn Simulation Multi-Conference, 2016, pp. 20–24.
[13] L. Zhou, L. Zhang, L. Ren, Simulation model of dynamic service scheduling in cloud manufacturing, in: IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp. 4199–4204.
[14] L. Zhou, L. Zhang, L. Ren, Modelling and simulation of logistics service selection in cloud manufacturing, Procedia CIRP 72 (2018) 916–921.
[15] L. Zhou, L. Zhang, L. Ren, Simulation of production modes for cloud manufacturing enterprises, in: 4th International Conference on Universal Village (UV), 2018, pp. 1–5.
[16] F. Li, L. Zhang, Y. Laili, Multi-task scheduling based on qos evaluation in cloud manufacturing system, in: International Manufacturing Science and Engineering Conference, 2017.
[17] F. Li, L. Zhang, T.W. Liao, et al., Multi-objective optimisation of multi-task scheduling in cloud manufacturing, Int. J. Prod. Res. 57 (12) (2019) 3847–3863.
[18] F. Li, T.W. Liao, L. Zhang, Two-level multi-task scheduling in a cloud manufacturing environment, Robot. Comput. Integr. Manuf. 56 (2019) 127–139.
[19] S. Liu, L. Zhang, W. Zhang, et al., Game theory based multi-task scheduling of decentralized 3D printing services in cloud manufacturing, Neurocomputing 446 (2021) 74–85.
[20] Y. Hu, F. Zhu, L. Zhang, et al., Scheduling of manufacturers based on chaos optimization algorithm in cloud manufacturing, Robot. Comput. Integr. Manuf. 58 (2019) 13–20.
[21] R. Doriya, P. Chakraborty, G.C. Nandi, Robotic services in cloud computing paradigm, in: International Symposium on Cloud Services Computing, IEEE Computer Society, 2012, pp. 80–83.
[22] Z. Du, W. Yang, Design of a robot cloud center, in: Tenth International Symposium on Autonomous Decentralized Systems, IEEE, 2011, pp. 269–275.
[23] X. Xu, From cloud computing to cloud manufacturing, Robot. Comput. Integr. Manuf. 28 (1) (2012) 75–86, 2012.
[24] H. Yan, Q. Hua, Y. Wang, et al., Cloud robotics in smart manufacturing environments: challenges and countermeasures, Comput. Electr. Eng. 63 (2017) 56–65.
[25] Y. Zhao, Q. Liu, W. Xu, et al., Dynamic and unified modelling of sustainable manufacturing capability for industrial robots in cloud manufacturing, Int. J. Adv. Manuf. Technol. 93 (5) (2017) 2753–2771, 2017.
[26] Z. Zhang, X. Wang, X. Zhu, et al., Cloud manufacturing paradigm with ubiquitous robotic system for product customization, Robot. Comput. Integr. Manuf. 60 (2019) 12–22.
[27] L. Wang, R. Gao, I. Ragai, An integrated cyber-physical system for cloud manufacturing, International Manufacturing Science and Engineering Conference, Am. Soc. Mech. Engineers (2014).
[28] Z. Zhang, W. Xu, Q. Liu, et al., Dynamic manufacturing capability assessment of industrial robots based on feedback information in cloud manufacturing, International Manufacturing Science and Engineering Conference, Am. Soc. Mech. Engineers (2017) 50749.
[29] C. Yang, Y. Wang, S. Lan, et al., Cloud-edge-device collaboration mechanisms of deep learning models for smart robots in mass personalization, Robot. Comput. Integr. Manuf. (2022) 77.
[30] Y. Zhao, Y. Liu, et al., A framework for development of digital twin industrial robot production lines based on a mechatronics approach, Int. J. Model., Simul., Sci. Comput. (2022).
[31] C. Wang, L. Zhang, et al., Adaptive scheduling method for dynamic robotic cell based on pattern classification algorithm, Int. J. Model., Simul., Sci. Comput. 9 (05) (2018).
[32] V. Mnih, K. Kavukcuoglu, D. Silver, et al., Playing atari with deep reinforcement learning, Comput. Sci. (2013).
[33] H.V. Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, Comput. Sci. (2015).
[34] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, N. De Freitas, Dueling network architectures for deep reinforcement learning, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 1995–2003.
[35] T. Schaul, J. Quan, I. Antonoglou, et al., Prioritized experience replay, Comput. Sci. (2015).
[36] M. Hausknecht, P. Stone, Deep recurrent q-learning for partially observable MDPs, Comput. Sci. (2015).
[37] T.P. Lillicrap, J.J. Hunt, A. Pritzel, et al., Continuous control with deep reinforcement learning, Comput. Sci. (2015).
[38] V. Mnih, A.P. Badia, M. Mirza, et al., Asynchronous methods for deep reinforcement learning, Int. Conf. Mach. Learn., PMLR (2016) 1928, 1937.
[39] Y. Liu, L. Zhang, L. Wang, et al., A framework for scheduling in cloud manufacturing with deep reinforcement learning, in: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), IEEE, 2019.
[40] L. Zhou, L. Zhang, B. Horn, Deep reinforcement learning-based dynamic scheduling in smart manufacturing, Procedia CIRP 93 (2020) 383–388.
[41] H. Zhu, M. Li, Y. Tang, et al., A deep-reinforcement-learning-based optimization approach for real-time scheduling in cloud manufacturing, IEEE Access 8 (2020) 9987–9997.
[42] S. Yang, Z. Xu, Intelligent scheduling and reconfiguration via deep reinforcement learning in smart manufacturing, Int. J. Prod. Res. (2021) 1–18.
[43] H. Du, W. Xu, B. Yao, et al., Collaborative optimization of service scheduling for industrial cloud robotics based on knowledge sharing, Procedia CIRP 83 (2019) 132–138.
[44] K. Mei, Y. Fang, Multi-robotic disassembly line balancing using deep reinforcement learning, International Manufacturing Science and Engineering Conference, Am. Soc. Mech. Engineers (2021) 85079.
[45] Z. Yin, J. Liu, D. Wang, et al., Multi-AGV task allocation with attention based on deep reinforcement learning, Int. J. Pattern Recognit. Artif. Intell. (2022).
[46] L. Equeter, C. Letot, R. Serra, et al., Estimate of cutting tool lifespan through cox proportional hazards model, IFAC-PapersOnLine 49 (28) (2016) 238–243.
[47] A.R. Mashhadi, S. Behdad, Optimal sorting policies in remanufacturing systems: application of product life-cycle data in quality grading and end-of-use recovery, J. Manuf. Syst. 43 (2017) 15–24.
[48] X. Tang, K. Li, Z. Zeng, et al., A novel security-driven scheduling algorithm for precedence-constrained tasks in heterogeneous distributed systems, IEEE Trans. Comput. 60 (7) (2010) 1017–1029.