Received January 17, 2019, accepted February 11, 2019, date of current version April 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902846

Multi-Objective Workflow Scheduling With Deep-Q-Network-Based Multi-Agent Reinforcement Learning

YUANDOU WANG[1], HANG LIU[1], WANBO ZHENG[2], YUNNI XIA[1], (Senior Member, IEEE), YAWEN LI[1], PENG CHEN[3], KUNYIN GUO[1], AND HONG XIE[4]

[1]College of Computer Science, Chongqing University, Chongqing 400044, China
[2]Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China
[3]College of Computer Science, Sichuan University, Chengdu 610065, China
[4]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong 999077

Corresponding authors: Wanbo Zheng (zwanbo2001@163.com) and Yunni Xia (xiayunni@hotmail.com)

This work was supported in part by the NSFC under Grant 61472051, in part by the Fundamental Research Funds for the Central Universities under Project 106112014CDJZR185503 and Project CDJZR12180012, in part by the Science Foundation of Chongqing under Grant cstc2014jcyjA40010 and Grant cstc2014jcyjA90027, in part by the Chongqing Social Undertakings and Livelihood Security Science and Technology Innovation Project Special Program under Grant cstc2016shmszx90002, in part by the China Postdoctoral Science Foundation under Grant 2015M570770, in part by the Chongqing Postdoctoral Science special Foundation under Grant Xm2015078, in part by the Universities Sci-tech Achievements Transformation Project of Chongqing under Grant KJZH17104, in part by the Guangzhou Science and Technology Projects under Grant 201807010052 and Grant 201610010092, and in part by the Nansha Science and Technology Projects under Grant 2017GJ001.

ABSTRACT Cloud Computing provides an effective platform for executing large-scale and complex workflow applications with a pay-as-you-go model. Nevertheless, various challenges, especially its optimal scheduling for multiple conflicting objectives, are yet to be addressed properly. The existing multi-objective workflow scheduling approaches are still limited in many ways, e.g., encoding is restricted by prior experts' knowledge when handling a dynamic real-time problem, which strongly influences the performance of scheduling. In this paper, we apply a deep-Q-network model in a multi-agent reinforcement learning setting to guide the scheduling of multi-workflows over infrastructure-as-a-service clouds. To optimize multi-workflow completion time and user's cost, we consider a Markov game model, which takes the number of workflow applications and heterogeneous virtual machines as state input and the maximum completion time and cost as rewards. The game model is capable of seeking for correlated equilibrium between make-span and cost criteria without prior experts' knowledge and converges to the correlated equilibrium policy in a dynamic real-time environment. To validate our proposed approach, we conduct extensive case studies based on multiple well-known scientific workflow templates and Amazon EC2 cloud. The experimental results clearly suggest that our proposed approach outperforms traditional ones, e.g., non-dominated sorting genetic algorithm-II, multi-objective particle swarm optimization, and game-theoretic-based greedy algorithms, in terms of optimality of scheduling plans generated.

INDEX TERMS Multi-objective workflow scheduling, deep-Q-network (DQN), multi-agent reinforcement learning (MARL), infrastructure-as-a-service (IaaS) cloud, quality-of-service (QoS).

LIST OF ABBREVIATIONS

IaaS	Infrastructure-as-a-service
QoS	Quality-of-Service
DAG	Directed-Acyclic-Graph
DQN	Deep Q-network
MARL	Multi-agent Reinforcement Learning
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
MOPSO	Multi-objective Particle Swarm Optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Shuiguang Deng.

LIST OF SYMBOLS

N	The number of workflows
K	The number of bags of tasks
n_k	The number of tasks in the k^{th} bag of tasks
$n_{k,i}$	The i^{th} task of n_k
M	The number of Amazon EC2 instances
V_j	The j^{th} virtual machine of Amazon EC2 instances
p_j	The unit price of instance type V_j
$st_{k,i,j}$	The start time of $n_{k,i}$ executed by V_j
$rt_{k,i,j}$	The running time of $n_{k,i}$ executed by V_j

$FT(V_j, n_{k,i})$	The finish time of V_j when $n_{k,i}$ is executed
$x_{k,i,j}$	A Boolean variable indicating whether V_j is selected for $n_{i,k}$; 0, otherwise
I	The players or agents of game
S	The state space of the Markov game
$A, \mathcal{A}$	The action spaces, mixed actions of game
R	The rewards of players
P	The transition probability of the game
a_t	The action profile at date-t
a_i^t	The action of player-i of the game at date-t
a_{-i}^t	The action of the other players of the game at date-t
s^t	The state of the game at date-t
R_i	The rewards of player-i
π	The stationary policy
π_{s^t}	The distribution policy with state s at date-t
π^t	The distribution policy at date-t
δ^t	The discount factor at date-t
f	The selection mechanism

I. INTRODUCTION

Cloud Computing is emerging as a high performance computing environment with a large-scale, heterogeneous collection of autonomous systems and flexible computational architecture [1], [2]. It provides the tools and technologies to build data or computational intensive parallel applications with much more affordable prices compared to traditional parallel computing techniques. Hence, there has been an increasingly growth in the number of active research work in cloud computing such as scheduling, placement, energy management, privacy and policy, security [3]–[6], etc.

Workflow scheduling in cloud environment has recently drawn enormous attention thanks to its wide application in both scientific and economic areas [7]–[12]. A workflow is usually formulized as a Directed-Acyclic-Graph (DAG) with several n tasks that satisfy the precedent constraints. Scheduling workflows in clouds is referred to as matching tasks onto m supporting computational resources, i.e., virtual machines (VMs) created on IaaS clouds. For multi-objective scheduling, objectives can sometimes be conflicting. E.g., for execution time minimization, fast VMs are more preferable than slow ones. However, fast VMs are usually more expensive and thus execution time minimization may contradict the cost reduction objective. It is widely acknowledged as well that to schedule multi-task workflow on distributed platforms is an NP-hard problem. It is therefore extremely time-consuming to yield optimal schedules through traversal-based algorithms. Fortunately, heuristic and meta-heuristic algorithms with polynomial complexity are able to produce approximate or near optimal solutions of schedules at the cost of acceptable optimality loss [13]. Good examples of such algorithms are multi-objective particle swarm optimization (MOPSO) [14] and non-dominated sorting genetic algorithm-II (NSGA-II) [15]. Although these algorithms provide satisfactory solutions, they require a lot of prior experts' knowledge and human intervention, usually in terms of encoding schemes. It is noticed by various contributions [16]–[19] as well that game-theoretic models and approaches are highly capable of dealing with the cloud-based workflow scheduling problems.

Recently, as novel machine learning algorithms are becoming increasingly versatile and powerful, considerable research efforts are paid to using reinforcement learning (RL) and Q-learning-based algorithms [20]–[23] in finding near-optimal workflow scheduling solutions. Nevertheless, most existing contributions focused on single-objective workflow scheduling with service-of-level (SLA) agreement constraints. Although there exist various multi-agent reinforcement learning (MARL) models and methods for multi-robot control, decentralized network routing, distributed load-balancing, electronic auctions, and traffic control problems, MARL-based workflow scheduling methods are still non-existent.

Based on above observations, in this work, we formulate the scheduling problem into a discrete-events and multi-criteria-interaction Markov game model and propose a multi-agent Deep-Q-network (DQN) algorithm with reinforcement learning for multi-objective workflow scheduling aiming at optimizing both workflow completion time and cost. The DQN agents are trained in a multi-agent reinforcement learning (MARL) environment and fed with data from legacy system such as heuristics in neural networks. We consider each DQN agent observes all the other agents' actions and rewards and selects its own joint distribution action along with environment updates. The resulting workflow scheduling plans are generated through a self-learning and self-optimizing manner. Our proposed approach are featured by the following strengths: 1) Agents can be trained for workflows with varying types of process models and heterogeneous VMs with varying resource configurations; and 2) The destination scheduling plans can be obtained without human intervention or prior expert's knowledge. We conduct extensive case studies with multiple scientific workflow templates over simulation tests using real-world third-party IaaS cloud data. The experimental results clearly suggest that our proposed approach outperforms traditional ones in terms of both make-span and cost optimization.

II. RELATED WORK

It is widely known that to schedule multi-task workflow on distributed platforms is an NP-hard problem. It is therefore extremely time-consuming to yield optimal schedules through traversal-based algorithms. Fortunately, heuristic and meta-heuristic strategies with polynomial complexity are capable of producing approximate or near optimal solutions at the cost of acceptable optimality loss. E.g., Kaur *et al.* [24] proposed a multi-objective bacteria foraging optimization algorithm (MOBFOA) by modifying the original BFOA and considering Pareto-optimal fronts. They aimed at minimizing flow-time, make-span, and resource-usage cost. Zhang *et al.* [25] presented a bi-objective genetic algorithm (BOGA) capable of optimizing both energy savings and workflow reliability and obtaining near-optimal Pareto fronts. Casas *et al.* [26] presented an enhanced genetic Algorithm with Efficient Tune-In (GA-ETI) for scientific applications in cloud systems. It is capable of optimizing both workflow make-span and cost. Verma *et al.* [27] presented a non-dominated-sorting-based Hybrid Particle-Swarm-Optimization (HPSO) algorithm for workflow scheduling, which is capable of optimizing both execution time and cost. Zhou *et al.* [28] proposed a fuzzy dominance sort based heterogeneous earliest-finish-time (FDHEFT) algorithm capable of minimizing both cost and make-span of workflows deployed on IaaS clouds. However, these approaches are significantly restricted by prior expert's knowledge from static global point of view, which cannot appropriately describe the dynamic process of workflow scheduling.

Recently, game-theoretic and Reinforcement learning (RL) models and methodologies are widely applied to the multi-constraint process scheduling problems [29]–[35]. It is believed the equilibrium concept in game theories and multi-agent training methods are highly potent in dealing with multi-constraint and multi-objective optimization problems. E.g., Duan *et al.* [18] proposed a sequential cooperative game algorithm for cost and make-span optimization while fulfilling storage constraints for large-scale workflow scheduling. Cui *et al.* [22] provided a reinforcement-learning-based approach for multi-workflow scheduling with multiple priorities submitted at different times in cloud environment. Iranpour *et al.* [17] proposed a distributed load-balancing and admission-control algorithm based on a fuzzy game-theoretic model for large-scale SaaS clouds. Wu *et al.* [20] proposed an improved Q-learning algorithm with weighted fitness value function for optimization of completion time and load balancing in cloud environment.

III. SYSTEM MODEL

A scientific workflow is represented by a Directed Acyclic Graph (DAG) $W = (T, E)$, where $T = \{t_1, t_2, \ldots, t_n\}$. is a set of n tasks $\{t_1, t_2, \ldots, t_n\}$., E is a set of precedence dependencies. Each task t_i represents an individual application with a certain task running time rt_i on an instance. A precedence dependency $e_{ij} = (t_i, t_j)$ indicates that t_j starts only after t_i is accomplished and the data from t_i are received. The source and destination of a dependency e_{ij} are called the parent and the child task, respectively. The workflow starts and concludes by the entry and exit tasks, respectively. A dummy entry/exit task with zero execution time can be added as a sole entry/exit one if the original workflow has multiple entry/exit ones rather than a single one.

In this work, we consider IaaS clouds as the supporting platforms of multiple workflows. IaaS clouds offer numerous heterogeneous virtual machines with varying resource and pricing configurations for executing workflows tasks. The optimization problem can be formulated as follows,

$$Minf_1 = makespan = max\{FT(V_j, n_{k,i}) * x_{k,i,j}\} \tag{1}$$

$$Minf_2 = cost = \sum_{k=1}^{M} FT(V_j, n_{k,i}) * x_{k,i,j} * p_j \tag{2}$$

subject to,

$$i \in [1, n_k], \quad j \in [1, M], \ k \in [1, K] \tag{3}$$

$$FT(V_j, n_{k,i}) = st_{k,i,j} + rt_{k,i,j}, \quad FT(V_j, n_{k,i}) \geq 0 \tag{4}$$

where f_1, f_2 illustrate the two quantitative objectives, i.e., the make-span and user's cost, respectively. And p_j is the unit price of each virtual machine V_j. The boolean indicator $x_{k,i,j}$ equals 1 when task $n_{k,i}$ is allocated to virtual machine V_j; otherwise $x_{k,i,j} = 0$. $FT(V_j, n_{k,i})$ is the finish time when V_j mapping task $n_{k,i}$, in terms of start time $st_{k,i,j}$ and runtime $rt_{k,i,j}$.

To solve the above formulation, we consider a self-adaptive DQN-based MARL framework shown as Figure 1, which is capable of generating real-time workflow scheduling plans. The two optimization objectives are abstracted as two DQN agents, which are trained through self-adaptive process built upon a stochastic Markov game. The environment is modeled as a set of states and actions can be performed to control the workflow scheduling system's state.

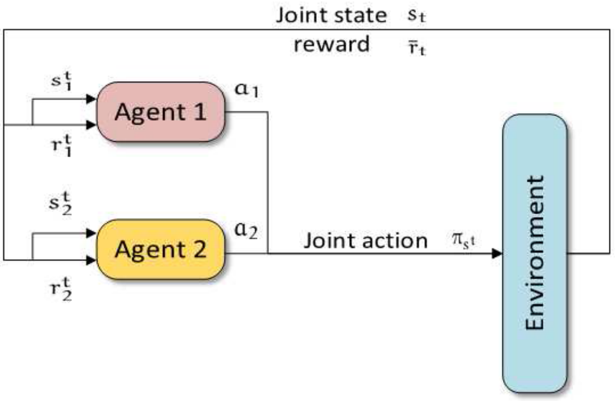


FIGURE 1. Overview of DQN-based MARL framework for workflow scheduling.

IV. METHODS: APPLICATION OF MARL TO WORKFLOW SCHEDULING

A. WORKFLOW SCHDEULING AS MARKOV GAME

Markov games can been seen as an extension of Markov Decision Processes (MDPs) to the multi-agent case, in which joint actions π_{s^t} are the result of multiple agents choosing an [36].

Definition 1: A(finite, discounted) **Markov game** is a tuple $\Gamma^{\delta} = (i \in I, S, A, R, P)$ in which [37],

- *I is a finite set of players or agents.*
- *S is a finite set of states.*

- $A = \prod_{i \in I, s \in S} A_i(s)$, *where* $A_i(s)$ *is player i's finite set of pure actions at state s; we define* $A(s) \equiv \prod_{i \in I} A_i s$ *and* $A_{-i} = \prod_{j \neq i} A_j(s)$, *so that* $A(s) = A_{-i}(s) \times A_i(s)$; *we write* $a = (a_{-i}, a_i) \in A(s)$ *to distinguish player i, with* $a_i \in A_i(s)$ *and* $a_{-i} \in A_{-i}(s)$; *we also define* $\mathcal{A} = \bigcup_{s \in S} \bigcup_{a \in A(s)} \{(s, a)\}$, *the set of state-action pairs.*
- *P is a system of transition probabilities, i.e., for all* $s \in S$, $a \in A(s)$, $P[s'|s, a] \geq 0$ *and* $\sum_{s' \in S} P[s'|s, a] = 1$; *we interpret* $P[s'|s, a]$ *as the probability that next state is* s' *given that the current state is s and the current action profile is a.*
- $R : \mathcal{A} \rightarrow [\alpha, \beta]^I$, *where* $R_i(s, a) \in [\alpha, \beta]$ *is player i's reward at state s and at action profile* $a \in A(s)$.
- $\delta \in [0, 1]$ *is a discounted factor.*

We consider the workflow scheduling process to be a Markov game with scheduling goals, i.e., make-span and cost, as two agents. It is assumed that each agent observes each other's actions and rewards. Then they select the joint distribution π_{s^t}, i.e., the combination of choices of all agents. Each agent further decides an action a_i^t and the resulting pure action profile $a^t = (a_1^t, \ldots, a_I^t)$ is performed. Based on the current state and the action profile, each agent earns reward $R_i(s^t, a^t)$ and the system evolves to state s^{t+1} with the transition probability $P[s^{t+1}|s^t, a^t]$. The above process is repeated at time $t + 1$. A state of the space S is characterized by the currently available VMs and immediately succeeding tasks of those which are mapped to destination VMs for execution in the previous state. The action space A consists of the mapping probability of certain task being mapped into a certain VM. The reward $R : A \rightarrow \mathbb{R}^n$ are derived from (1) and (2).

Note that the performance of scheduling is directly influenced by the reward mechanisms along with the interactions among agents. There may exist multiple equilibria with multiple values. In order to resolve this problem, we introduce an utilitarian selection mechanism $f = max_{\pi_s \in \Delta(A(s))} \sum_{j \in I} Q_j(s, a)$, which donates maximize the sum of all agents' rewards in each state. Usually, the equilibrium policies in a Markov game are solutions of problem with stable results. Instead of Nash equilibrium, we consider a correlated equilibrium with increased generality. It allows for dependencies among agents' strategies, that is a joint distribution over actions from which no agent is motivated to deviate unilaterally. The solutions of the workflow scheduling problem are thus correlated equilibria, where agents are allowed to select actions according to a stationary policy $\pi \in \prod_{s \in S} \Delta(A(s))$.

Definition 2: Given a Markov game, a stationary policy π is a correlated equilibrium if for all agent $i \in I$, for all $s \in S$, for all $a_i, a'_{-i} \in A_i(s)$,

$$\sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i) Q_i^{\pi}(s, (a_{-i}, a_i)) \geq \sum_{a_{-i} \in A_{-i}(s)} \pi_s(a_{-i}, a_i) Q_i^{\pi}(s, (a_{-i}, a'_i)) \quad (5)$$

Algorithm 1 DQN-Based MARL Method

```
Input: game $\Gamma$, selection mechanism $f$
Output: Q-values $Q$, stationary policy $\pi^*$, reward $r$
Initialize replay memory $D$, action-value function $Q$
  with random weights $\theta$;
Initialize state $s$, action profile $a$;
observe initial state $S$;
while not at max_episode do
    if with probability $\varepsilon$ then
        select a random action $a$;
    else
        select $a \in f$;
    carry out action $a$;
    observe reward $r$ and new state $s'$;
    store experience $< s, a, r, s' >$ in replay memory $D$;
    sample random transitions $< ss, aa, rr, ss' >$ from
      replay memory $D$;
    calculate target for each minibatch transition;
    if $ss'$ is terminal state then
        $tt = rr$;
    else
        $tt = rr + \delta\ max_{a'} Q(ss', aa')$;
    train the Q-network using $(tt - Q(ss, aa))^2$ as loss;
    $s = s'$
return Q-values $Q$, action profile $a$, reward $r$
```

That is, in state s, when it is recommended that agent i play a_i, it prefers to play a_i, because the expected utility of a_i is greater than or equal to the expected utility of a'_i, for all a'_i.

B. DQN-BASED MARL IN WORKFLOW SCHEDULING

DQN [38] is a popular method in reinforcement learning. It learns the action-value function Q^* corresponding to the optimal policy by minimizing the loss,

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}[(Q^*(s, a|\theta) - y)^2] \quad (6)$$

$$y = r + \delta max_{a'} Q^*(s', a') \quad (7)$$

where y is a target Q function whose parameters are periodically updated with the most recent θ, which helps stabilize learning. Another crucial component of stabilizing DQN is the use of an experience replay buffer D containing tuples (s, a, r, s'). The agent determines its actions by using a neural network and mixing the output of the neural network with random actions to sample its training set. Usually, the agent trains the neural network in such a way that it predicts the cumulative, weighted rewards for all actions.

The optimal policy of a DQN-based agent not only interacts directly with the workflow environment, but also with the policies of the other agents as well. The iterative algorithm for computing global equilibrium policies based on local updates Q-values, policy at each state. Generally, Q-values are given at time t for all $i \in I$, for all $s \in S$, and for all $a \in A(s)$, namely $Q_i^t(s, a)$. To achieve a correlated equilibrium, each DQN

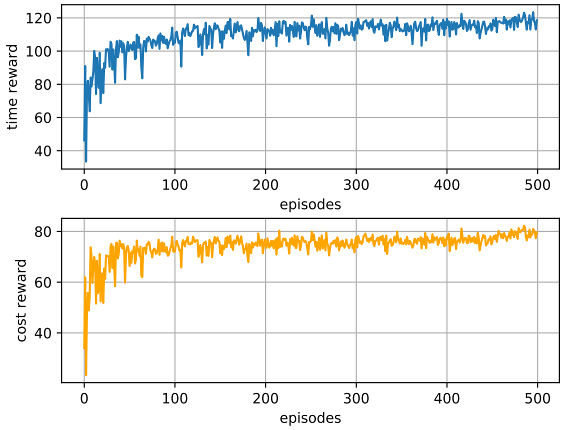


FIGURE 2. The convergence of DQN-based MARL framework.

agent learns about the correlated equilibrium strategy π^t, where $\pi_s^{t+1} \in f(Q^{t+1}(s))$, the DQN-based MARL algorithm is developed in **Algorithm 1**. Along with suitable reward mechanisms designed, the convergence of the DQN-based algorithm in multi-agent settings can be guaranteed. For the make-span agent, the reward mechanism is designed as,

$$\Re_1 = [\frac{ET_{k,i,j}(a) - (makespan' - makespan)}{ET_{k,i,j}(a)}]^3 \tag{8}$$

similarly, the cost reward is designed as,

$$\Re_2 = [\frac{worest - ET_{k,i,j}(a) * p_j}{worest - best}]^3 \tag{9}$$

TABLE 1. Units for price of Amazon EC2 instances.

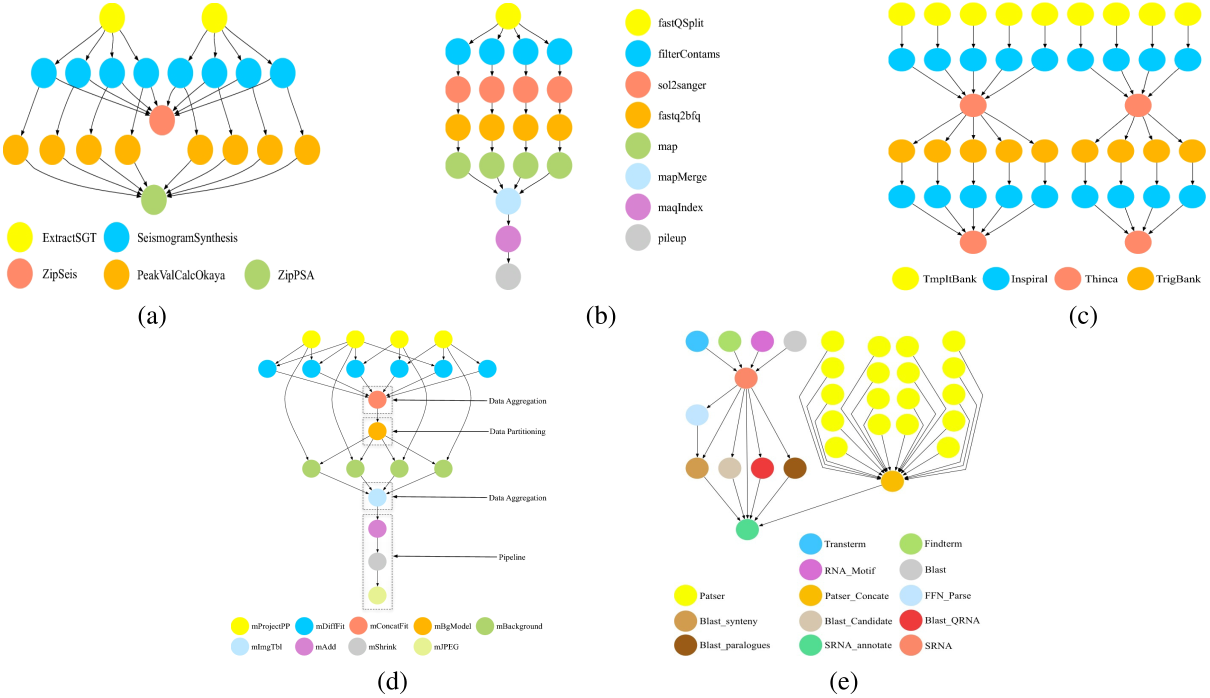
vTypes	vCPU	Memory	Availability Zone	Price (USD$/hr)
$t3.medium$	2	8	$us-east-2a$	0.0418
$t3.large$	2	8	$us-east-2c$	0.0835
$c5.large$	4	16	$us-east-2b$	0.0850
$m5.large$	4	16	$us-east-2a$	0.0960
$c5n.large$	8	32	$us-east-2c$	0.1080
$r5a.large$	8	32	$us-east-2b$	0.1260
$a1.4xlarge$	8	32	$us-east-2a$	0.4080

where (8) indicates a lower increase of make-span is more preferable. Similarly, (9) indicates that a lower increase of cost is more preferable. Figure 2 demonstrates the convergence of our proposed approach with respect to make-span and cost.

V. EXPERIMENTS, RESULTS AND DISCUSSION

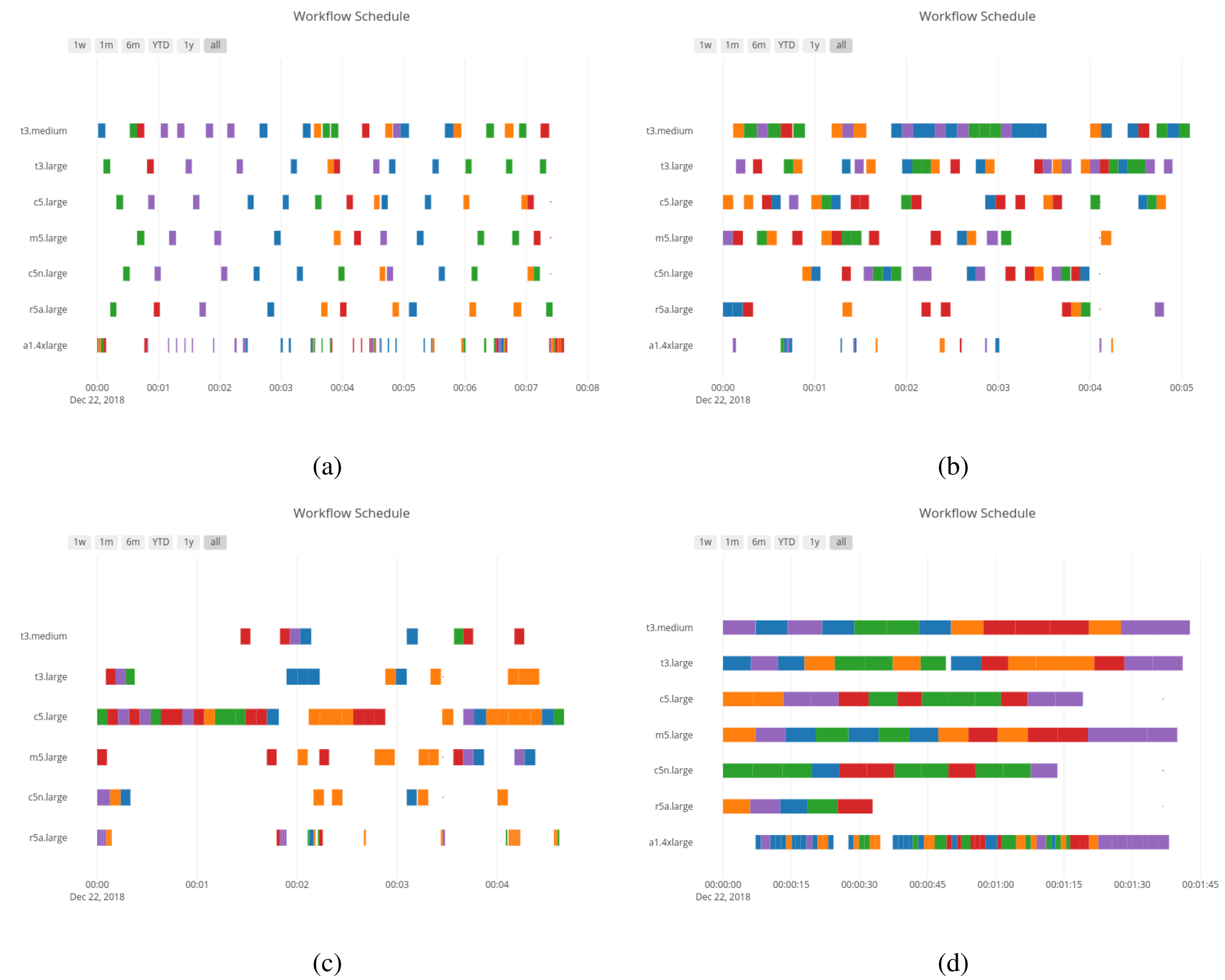
For the model validation purpose, we conduct extensive case studies based on multiple well-known scientific workflow templates shown in Figure 3 and real-world third-party commercial clouds, i.e., the Amazon EC2 shown in Table 1.

We consider different types of tasks, namely AES, LZMA, JPEG, Canny and Lua workloads that simulate task execution scenarios, are performed by the workflow templates shown in Figure 3. As shown in Table 2 from Geekbench [39], the performances of tasks are varying based on the type of supporting VMs from Amazon EC2. For the comparison reason, we consider MOPSO [14], NSGA-II [15] and



(a) (b) (c) (d) (e)

FIGURE 3. Overview of five workflow templates. (a) CyberShake, (b) Epigenomics, (c) Inspiral, (d) Montage, and (e) Sipht.



(a)

(b)

(c)

(d)

FIGURE 4. Illustration of workflow scheduling with 138 tasks over NSGA-II, MOPSO, GTBGA and DQN-based MARL algorithms. (a) GTBGA; (b) NSGA-II; (c) MOPSO and (d) DQN-based MARL algorithm.

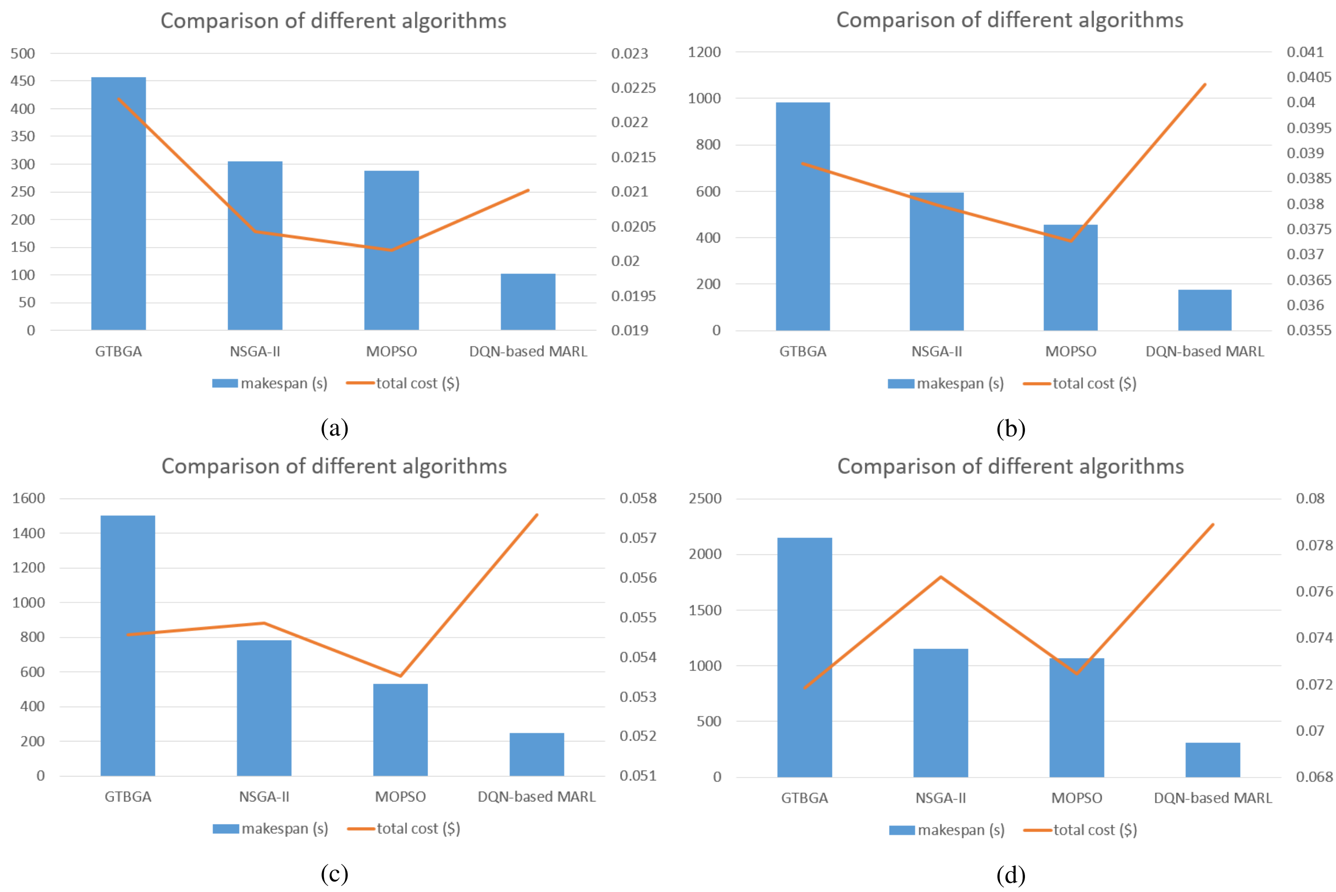
TABLE 2. Units for multi-core performance scores of instances.

vTypes	AES	LZMA	JPEG	Canny	Lua	Average
$t3.medium$	4217	4243	4228	3956	3537	4065
$t3.large$	4515	4905	5386	5080	4875	5035
$c5.large$	4525	4791	5526	5161	4936	4521
$m5.large$	4184	4490	4940	4678	4532	4575
$c5n.large$	4600	4967	5487	5122	4982	5105
$r5a.large$	5019	4768	4777	4530	3899	4549
$a1.4xlarge$	11489	25676	33355	14516	23089	16861

game-theoretic based greedy algorithm(GTBGA) [19] as the baseline algorithms.

Figure 4 illustrates the scheduling results with relatively small number of total tasks of five workflows, in terms of Gantt charts, of different algorithms. It can be seen that our proposed algorithm outperforms baseline algorithms in terms of make-span. Intuitively, the advantage of our algorithm is achieved due to the fact that our algorithm leaves less inter-task dwelling time and squeezes to more exploit the underlying parallelism provisioned by the EC2 platforms. In contrast, baseline algorithms tend to follow the topological constraint of workflows first and hesitate to fully exploit the potential parallelism.

Figure 5 shows the resulting comparison of make-span and cost achieved by different algorithms based on the scheduling plans shown in Figure 4. When taking into account both make-span and cost metrics, the performance of MOPSO algorithm apparently outweighs that of GTBGA and NSGA-II. Although MOPSO algorithm is cheaper than our proposed algorithm, our proposed algorithm clearly beats baseline ones in terms of make-span, and our algorithm is much cheaper than the GTBGA and only 2.899%, 4.303% more expensive than NSGA-II and MOPSO, respectively when task size is 138. According to Figure 6, we can clearly see that there is little difference between the baseline ones and our proposed method in terms of total cost for each kind of task size.



(a)

(b)

(c)

(d)

FIGURE 5. Illustration of comparisons of baseline algorithms and DQN-based MARL method with four different task sizes. (a) 138; (b) 252; (c) 358 and (d) 497.

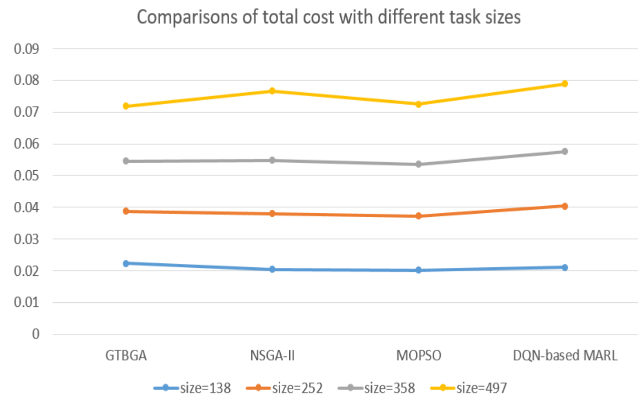


FIGURE 6. Comparisons of baseline algorithms and DQN-based MARL method with four different task sizes.

VI. CONCLUSION

This paper targets at the problem of multi-objective workflow scheduling in the heterogeneous IaaS cloud environment. We model the multi-objective workflow scheduling as a stochastic Markov game and develop a decentralized DQN-based MARL framework that is capable of obtaining correlated equilibrium solutions of workflow scheduling. The proposed DQN-based MARL framework is featured by the combination the traditional DQN algorithm for reinforcement learning and the novel model of cooperative and correlated equilibrium. We conduct extensive case studies based on Amazon EC2 and multiple scientific workflow templates and show that our proposed method outperforms the baseline algorithms such as NSGA-II, MOPSO and GTBGA.

As future work, we plan to consider more QoS metrics, such as reliability, security, load-balancing, etc. and introduce corresponding algorithms for on-the-fly scheduling for cross-organizational workflows. our proposed method relies on knowledge of QoS data of all tasks and candidate cloud servers. However, in practice it would be too expensive and time-consuming to collect such data at run-time. We thus intend to introduce large-scale-sparse-matrices-analysis models [40], [41] for QoS prediction when historical QoS data is insufficient.

ACKNOWLEDGMENT

(Yuandou Wang and Hang Liu contributed equally on this work.)

REFERENCES

[1] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, ''Stochastic modeling and performance analysis of migration-enabled and error-prone clouds,'' *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 495–504, Apr. 2015.

[2] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, ''A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters,'' *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 73–83, Jan. 2015.

[3] Y. Yin, Y. Xu, W. Xu, M. Gao, L. Yu, and Y. Pei, ''Collaborative service selection via ensemble learning in mixed mobile network environments,'' *Entropy*, vol. 19, no. 7, p. 358, 2017.

[4] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, ''Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing,'' *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1317–1332, May 2018.

[5] Y. Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, ''Network location-aware service recommendation with random walk in cyber-physical systems,'' *Sensors*, vol. 17, no. 9, p. 2059, 2017.

[6] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, ''iprivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1005–1016, 2017.

[7] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, ''A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing,'' *Future Gener. Comput. Syst.*, vol. 83, pp. 14–26, 2018.

[8] Q. Peng, M. Zhou, Q. He, Y. Xia, C. Wu, and S. Deng, ''Multi-objective optimization for location prediction of mobile devices in sensor-based applications,'' *IEEE Access*, vol. 6, pp. 77123–77132, 2018.

[9] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, ''Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds,'' *IEEE Access*, vol. 6, pp. 61488–61502, 2018.

[10] R. Xu *et al.*, ''Asufficient and necessary temporal violation handling point selection strategy in cloud workflow,'' *Future Gener. Comput. Syst.*, vol. 86, pp. 464–479, 2018.

[11] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, ''Computation offloading for service workflow in mobile cloud computing,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, pp. 3317–3329, Dec. 2015.

[12] J. Yu, D. Tao, M. Wang, and Y. Rui, ''Learning to rank using user clicks and visual features for image retrieval,'' *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 767–779, 2015.

[13] D. Nasonov, A. Visheratin, N. Butakov, N. Shindyapina, M. Melnik, and A. Boukhanovsky, ''Hybrid evolutionary workflow scheduling algorithm for dynamic heterogeneous distributed computational environment,'' *J. Appl. Logic*, vol. 24, pp. 50–61, 2017.

[14] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, ''Handling multiple objectives with particle swarm optimization,'' *IEEE Trans. Evol. Comput.*, vol. 8, pp. 256–279, Jun. 2004.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ''A fast and elitist multiobjective genetic algorithm: NSGA-II,'' *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, Apr. 2002.

[16] Y. Wang, J. Jiang, Y. Xia, Q. Wu, X. Luo, and Q. Zhu, ''A multi-stage dynamic game-theoretic approach for multi-workflow scheduling on heterogeneous virtual machines from multiple infrastructure-as-a-service clouds,'' in *Services Computing* (Lecture Notes in Computer Science), vol. 10969. Springer, 2018, pp. 137–152.

[17] E. Iranpour and S. Sharifian, ''A distributed load balancing and admission control algorithm based on Fuzzy type-2 and Game theory for large-scale SaaS cloud architectures,'' *Future Gener. Comput. Syst.*, vol. 86, pp. 81–98, Sep. 2018.

[18] R. Duan, R. Prodan, and X. Li, ''Multi-objective game theoretic scheduling of bag-of-tasks workflows on hybrid clouds,'' *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 29–42, 2014.

[19] L. Wu and Y. Wang, ''Scheduling multi-workflows over heterogeneous virtual machines with a multi-stage dynamic game-theoretic approach,'' *Int. J. Web Services Res.*, vol. 15, pp. 82–96, Oct. 2018.

[20] W. Jiahao, P. Zhiping, C. Delong, L. Qirui, and H. Jieguang, ''A multi-object optimization cloud workflow scheduling algorithm based on reinforcement learning,'' in *Intelligent Computing Theories and Application*. Cham, Switzerland: Springer, Aug. 2018, pp. 550–559.

[21] Y. Wei, D. Kudenko, S. Liu, L. Pan, L. Wu, and X. Meng, ''A reinforcement learning based workflow application scheduling approach in dynamic cloud environment,'' in *Collaborative Computing: Networking, Applications and Worksharing*. Cham, Switzerland: Springer, Dec. 2018, pp. 120–131.

[22] D. Cui, W. Ke, Z. Peng, and J. Zuo, ''Multiple DAGs workflow scheduling algorithm based on reinforcement learning in cloud computing,'' in *Computational Intelligence and Intelligent Systems*. Singapore, Springer, Nov. 2016, pp. 305–311.

[23] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek, ''Optimization of global production scheduling with deep reinforcement learning,'' *Procedia CIRP*, vol. 72, pp. 1264–1269, Jan. 2018.

[24] M. Kaur and S. Kadam, ''A novel multi-objective bacteria foraging optimization algorithm (MOBFOA) for multi-objective scheduling,'' *Appl. Soft Comput. J.*, vol. 66, pp. 183–195, 2018.

[25] L. Zhang, K. Li, C. Li, and K. Li, ''Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems,'' *Inf. Sci.*, vol. 379, pp. 241–256, Feb. 2017.

[26] I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, ''GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments,'' *J. Comput. Sci.*, vol. 26, pp. 318–331, May 2018.

[27] A. Verma and S. Kaushal, ''A hybrid multi-objective particle swarm optimization for scientific workflow scheduling,'' *Parallel Comput.*, vol. 62, pp. 1–19, Feb. 2017.

[28] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei, and S. Hu, ''Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT,'' *Future Gener. Comput. Syst.*, vol. 93, pp. 278–289, Apr. 2019.

[29] D. P. Bertsekas, ''Feature-based aggregation and deep reinforcement learning: A survey and some new implementations,'' *IEEE/ACM Trans. Audio, Speech, Language Process.*, pp. 1–31, 2018.

[30] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, ''Resource management with deep reinforcement learning,'' in *Proc. 15th ACM Workshop Hot Topics Netw. (HotNets)*, 2016, pp. 50–56.

[31] L. Xue, C. Sun, D. Wunsch, Y. Zhou, and F. Yu, ''An adaptive strategy via reinforcement learning for the prisoner's dilemma game,'' *IEEE/CAA J. Autom. Sinica*, vol. 5, pp. 301–310, Jan. 2018.

[32] Y. Zhan, H. B. Ammar, and M. E. Taylor, ''Theoretically-grounded policy advice from multiple teachers in reinforcement learning settings with applications to negative transfer,'' in *Proc. IJCAI Int. Joint Conf. Artif. Intell.*, 2016, pp. 2315–2321.

[33] H. Wang, T. Huang, X. Liao, H. Abu-Rub, and G. Chen, ''Reinforcement learning for constrained energy trading games with incomplete information,'' *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3404–3416, Oct. 2017.

[34] L. Zheng, J. Yang, H. Cai, W. Zhang, J. Wang, and Y. Yu, ''MAgent: A many-agent reinforcement learning platform for artificial collective intelligence,'' pp. 1–2, 2017.

[35] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, ''Multi-agent actor-critic for mixed cooperative-competitive environments,'' in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 6379–6390.

[36] S. Kapoor, ''Multi-agent reinforcement learning: A report on challenges and approaches,'' pp. 1–24, 2018.

[37] A. Greenwald, K. Hall, and R. Serrano, ''Correlated Q-learning,'' in *Proc. 20th Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 242–249.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, ''Playing atari with deep reinforcement learning,'' *CoRR*, vol. abs/1312.5602, 2013.

[39] t. f. e. From Wikipedia. *Geekbench.com*. Accessed: Jan. 15, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Geekbench

[40] Y. Yin, L. Chen, Y. Xu, and J. Wan, ''Location-aware service recommendation with enhanced probabilistic matrix factorization,'' *IEEE Access*, vol. 6, pp. 62815–62825, 2018.

[41] X. Fu, K. Yue, L. Liu, Y. Feng, and L. Liu, ''Reputation measurement for online services based on dominance relationships,'' *IEEE Trans. Services Comput.*, to be published.

YUANDOU WANG received the B.S. degree in computer science and technology from Chongqing University, Chongqing, China, in 2016, where she is currently pursuing the M.S. degree in computer science and technology with the College of Computer Science. Her current research interests include cloud computing, service computing, QoS engineering, and game theory.

HANG LIU received the B.S. degree in network engineering from Chongqing University, Chongqing, China, in 2018, where he is currently pursuing the master's degree in computer technology. His current research interests include edge computing, game theory, and workflow scheduling.

WANBO ZHENG received the B.S. degree in electrical and information engineering from the Chongqing University of Technology, China, in 2005, the M.S. degree in mining engineering from the China Coal Research Institute, China, in 2009, and the Ph.D. degree in software engineering from Chongqing University, China, in 2017. Since 2019, he has been an Associate Research Fellow with the Faculty of Science, Kunming University of Science and Technology. His current research interests include cloud computing, service computing, heterogeneous cloud services, and scientific workflows.

YUNNI XIA (SM'14) received the B.S. degree in computer science from Chongqing University, China, in 2003, and the Ph.D. degree in computer science from Peking University, China, in 2008. Since 2008, he has been a Professor with the School of Computer Science, Chongqing University. He has authored or co-authored over 50 research publications. His research interests are in Petri nets, software quality, performance evaluation, and cloud computing system dependability.

YAWEN LI is currently pursuing the B.S. degree in computer science and technology from Chongqing University, Chongqing, China. Her research interest includes cloud computing.

PENG CHEN received the B.Sc. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2001, and the M.Sc. degree in computer science from Peking University, Beijing, China, in 2004. He is currently pursuing the Ph.D. degree with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu. His research interests include blind signal processing and neural networks.

KUNYIN GUO received the B.S. degree in network engineering from Chongqing University, China, 2012, where he is currently pursuing the Ph.D. degree with the College of Computer Science. His research interests include cloud computing, fault-tolerant, and optimization problem.

HONG XIE received the B.Eng. degree from the School of Computer Science and Technology, University of Science and Technology of China, in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His advisor is Prof. J. C. S. Lui. His research interests include network economics, data analytics, stochastic modeling, crowdsourcing, and online social networks.