Cooperative Resource Allocation Based on Soft Actor–Critic With Data Augmentation in Cellular Network

Yunhui Qin, Zhongshan Zhang, *Senior Member, IEEE*, Wei Huangfu, *Member, IEEE*, Haijun Zhang, *Senior Member, IEEE*, and Keping Long, *Senior Member, IEEE*

***Abstract*—This letter investigates the cooperative resource allocation of cellular networks with simultaneous wireless information and power transfer in the time-varying channel environment. The soft actor-critic (SAC) algorithm is exploited to tackle the optimization problem which aims to find a feasible resource allocation policy to maximize the data rate and system fairness while minimizing the channel switching penalty. Considering the costly agent-to-environment interactions and the restricted empirical dataset of the SAC algorithm, this letter explores the permutation equivalence of the optimization objective, and designs two data augmentation schemes for the experience replay buffer of SAC. The cumulative discount reward shows that data augmentation assisted algorithms outperform the baseline in the learning speed. The simulation results referring to the average data rate and system fairness show that the proposed schemes benefit to the training model and effectively improve the performance of algorithms.**

***Index Terms*—Cooperative resource allocation, deep reinforcement learning, soft actor-critic.**

I. Introduction

With the ubiquitous service requirements and the ever-increasing population of users, it is challenging for the limited radio resources to satisfy various wireless services in the future cellular networks. Resource allocation of the cellular networks with simultaneous wireless information and power transfer (SWIPT) has attracted significant attention to relieve the contradiction between various service demands and constrained energy of nodes [1], [2], [3], [4]. Most of the existing research mainly focuses on one-time slot and is even infeasible in uncertain and dynamic network environments.

Recently, reinforcement learning (RL) algorithms for resource allocation have been regarded as promising techniques in the future cellular network [5], [6], [7], [8], [9]. In [6], the adaptive rate and energy harvesting of the multiple-input single-output SWIPT system were analyzed and an effective RL method was proposed to maximize the throughput under the energy constraint. In [7], the outage probability of wireless networks with energy harvest capability was explored and a novel Q-learning algorithm was proposed to dynamically allocate channel resources. In [8], the radio resource management in non-orthogonal multiple access networks was studied and the semi-supervised learning and deep neural network were proposed to maximize the energy efficiency of the system. In [9], the joint resource allocation for the multi-carrier non-orthogonal multiple access with SWIPT was studied and a deep learning method was proposed to minimize the total transmit power. Moreover, the off-policy deep RL algorithms, such as deep deterministic policy gradient (DDPG) [10] and soft actor-critic (SAC) [11], have the advantages of decorrelation for samples. Their experience replay mechanism randomizes the dataset to break correlations, enables agents to learn from experience, and further determines the optimal policy. In [10], the real-time resource management for the unmanned aerial vehicles assisted vehicular networks is analyzed, and DDPG is utilized to maximize offloading while meeting the quality of service demands. However, the collection of the empirical datasets for replay buffers is usually time-consuming and costly, which has a negative effect on the training speed of algorithms. It is challenging to improve the training speed and the availability of an off-policy algorithm.

This letter investigates the cooperative resource allocation in the SWIPT-enabled cellular network, where multi-mobile users with energy harvesting capability are served by base stations in the time-varying channel conditions. Inspired by existing studies [11], [12], we exploit the SAC algorithm to achieve cooperative resource allocation in the considered scenario. Considering the requirements of the data rate, the system fairness and the channel switching penalty term, the mixed integer non-linear optimization objective is formulated. In order to find a feasible resource allocation policy, we analyze the equivalent permutation of the optimization objective based on symmetric group and transform it into the new optimization problem. Inspired by data augmentation [12] for RL, we accordingly propose two schemes including i) the random and ii) the adaptive for the experience replay buffer of SAC. The cumulative discount reward shows that data augmentation-assisted SAC algorithms outperform the baseline in learning speed. Despite the proposed schemes increase the switching penalties, the experiment results still show that

Manuscript received 21 September 2022; accepted 30 November 2022. Date of publication 6 December 2022; date of current version 9 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071035. The associate editor coordinating the review of this article and approving it for publication was M. R. A. Khandaker. *(Corresponding author: Zhongshan Zhang.)*

Yunhui Qin is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the Yangtze Delta Region Academy, Beijing Institute of Technology, Jiaxing 314019, Zhejiang, China (e-mail: 7520220083@bit.edu.cn).

Zhongshan Zhang is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: zhangzs@bit.edu.cn).

Wei Huangfu, Haijun Zhang, and Keping Long are with the Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing 100083, China (e-mail: huangfuwei@ustb.edu.cn; haijunzhang@ieee.org; longkeping@ustb.edu.cn).

Digital Object Identifier 10.1109/LWC.2022.3227033

the proposed schemes outperform SAC in terms of the data rate and system fairness, especially for the adaptive scheme, which verifies the effectiveness and availability of the data augmentation schemes for supporting cooperative resource allocation in a time-varying SWIPT-enabled cellular network environment.

II. System Model and Problem Formulation

Consider a SWIPT-enabled cellular network with multi-mobile users to support information transfer and energy harvesting. Assume that there are *M* base stations and *N* users in the system. We denote the sets of all the base stations and users by $\mathcal{M}$ and $\mathcal{N}$, respectively. Moreover, there are *N* orthogonal channels preassigned to the users. A set of binary variables $\{x_{ij}^k\}_{\forall i \in \mathcal{M}, j \in \mathcal{N}}$ is defined to indicate the status of the channel occupying in the network. Here, if cellular user *j* occupies channel *k*, $x_{ij}^k = 1$; otherwise, $x_{ij}^k = 0$. Note that each cellular user is allowed to occupy only one channel, i.e.,$\sum_i x_{ij}^k = 1$.

In the time slot *t*, if $x_{ij,t}^k = 1$, the *j*th user will communicate with *i*th base station, and the transmit power is $p_{ij,t}$. The signal strength can be expressed as $p_{ij,t} g_{ij,t}^k$, where $g_{ij,t}^k$ denotes channel gain. If the signal strength is less than the predefined threshold, the user is re-allocated channel. Let h_t represent the channel switching number of times in adjacent time slots, $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ represent the occupied state vectors of channels in adjacent time slots, respectively. Thus, the number of channel switching can be expressed as $h_t = \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_1$, $t \geq 1$. In order to ensure the stability of the system, the frequent channel switching is usually avoided, that is to minimize $\sum_t h_t$.

In this letter, mobile users also harvest and store energy of signals in addition to information transmission. Suppose that the energy is stored at intervals of *z* time slots, the remaining energy of *j*th user can be written as

$$e_{j,t+1} = \min\Big\{\max\big\{e_{j,t} + \mathbf{1}(t\%z = 0)\Delta\mathrm{E}_{j,t} - p_{ij,t}\tau_{ij,t}, 0\big\}, e_j^{\max}\Big\} \tag{1}$$

where $\mathbf{1}(\cdot)$ is indicator function, $\Delta\mathrm{E}_{j,t}$ denotes the harvested energy, $\tau_{ij,t}$ is effective communication time during slot *t*, and $e_j^{\max}$ is the maximum energy constraint of *j*th user.

The data rate of *j*th user in time slot *t* is given by

$$c_{j,t} = \sum_{i=1}^{M}\sum_{k=1}^{K}\left(x_{ij,t}^k \tau_{ij,t} w \log(1 + \frac{p_{ij,t} g_{ij,t}^k}{I_{ij,t}})\right) \tag{2}$$

where $I_{ij,t} = o^2 + \sum_{i'=1}^{M}\sum_{j'=1}^{N} x_{i'j',t}^k p_{i'j',t} g_{i'j',t}^k$, $j' \neq j$, *w* denotes the allocated bandwidth and o^2 is the variance of additive white Gaussian noise. Thus, the total data rate can be expressed as $C_t = \sum_{j=1}^{N} c_{j,t}$.

To ensure the fairness of system, we introduce Jain's fairness index [13]. Here, the corresponding fairness index is

$$f_t = \frac{\left(\sum_{j=1}^{N} c_{j,t}\right)^2}{N\sum_{j=1}^{N} c_{j,t}^2} \tag{3}$$

where $f_t \in [0, 1]$, and the larger the fairness index, the fairer the system service quality.

Therefore, the optimization objective can be written as

$$\begin{aligned}(P0)\ \max_{\boldsymbol{x},\boldsymbol{p},\boldsymbol{\tau}}\ & \sum_{t=0}^{T}(y_1 C_t + y_2 f_t - y_3 h_t)\\ & \mathrm{C1}: x_{ij,t}^k \in \{0, 1\}\\ & \mathrm{C2}: 0 \leq p_{ij,t} \leq p^{max}\\ & \mathrm{C3}: 0 \leq \tau_{ij,t} \leq t\\ & \mathrm{C4}: i \in [1, M], j \in [1, N], k \in [1, K],\end{aligned} \tag{4}$$

where $\boldsymbol{x}$ is the status of channel occupying, $\boldsymbol{p}$ and $\boldsymbol{\tau}$ denote the allocated power vector and the effective communication time in every time slot, respectively. $y_i \geq 0$ denotes the proportion of each sub-objective in this problem.

In this letter, we aim to find a resource allocation policy that can 1) maximize the total data rate; 2) maximize the system fairness, and 3) minimize the channel switching penalty in every time slot. It is quite challenging to achieve all of these objectives, because on one hand, to provide the optimal fairness index and data rate, it is preferred to utilize the enumeration method such that the optimal channel and energy allocation policy can be obtained; on the other hand, to ensure system stability and minimize switching penalty, it is preferred to avoid frequent channel switching. Hence, it is difficult for the traditional method to solve this mixed integer non-linear and non-convex problem, and a feasible solution is supposed to be introduced.

III. Problem Solution

In this section, we first introduce the solution to the optimization problem. Then, we discuss the equivalent transformation of the optimization objective. Last, we propose two data augmentation schemes for the experience replay buffer.

A. The Problem Solution

The SAC algorithm, one of the off-policy deep RL methods, aims to determine the optimal policy that maximizes the long-term reward as well as strategy entropy. This algorithm consists of two major parts, i.e., the agent and environment. During step *t* in each episode, the agent first obtains the state s_t from environment and selects policy a_t from the action space. Then, the environment updates the current state to s_{t+1} and obtains the corresponding reward r_t. The experience tuple (s_t, a_t, r_t, s_{t+1}) is subsequently stored in experience replay buffer $\mathbf{D}$. This algorithm has better stability owing to its encouraging exploration [14].

The corresponding elements of SAC are defined as follows,

- The policy a_t is defined as

$$a_t = \{p_{11,t}, \ldots, p_{MN,t}; \tau_{11,t}, \ldots, \tau_{MN,t}\} \tag{5}$$

where $p_{ij,t}$ denotes the power allocated by *j*th user, and $\tau_{ij,t}$ is the effective communication time. Since decision variables take continuous values, it is a continuous resource allocation task.

- The state s_t is defined as

$$s_t = \{x_{11,t}^1, \ldots, x_{ij,t}^K; g_{11,t}^1, \ldots, g_{ij,t}^K; e_{1,t}, \ldots, e_{N,t}\} \tag{6}$$

where $x^k_{ij,t} \in \{0,1\}$ is channel occupying state, $g^k_{ij,t}$ is channel gain, and $e_{j,t}$ is the remaining energy of *j*th user.

- In this letter, the objective is to maximize (*P*0), thus the reward function can be written as

$$r_t = y_1 C_t + y_2 f_t - y_3 h_t \tag{7}$$

The SAC algorithm aims to maximize the long-term cumulative discount reward as well as strategy entropy, that is

$$\max \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1}[r_t(s_t,a_t) - \alpha \log \pi_\phi(a_t|s_t)]\right] \tag{8}$$

where $\gamma \in (0,1)$ denotes the discount factor, α is the temperature parameter, and π_ϕ denotes actor network.

The SAC algorithm mainly consists of two main critic networks with parameter vectors $\theta_i, i \in \{1,2\}$, two target critic networks with $\theta'_i, i \in \{1,2\}$ and one actor network with ϕ. According to the agent interacting with environment, the experience tuples are gradually generated and their numbers increase until more than enough be sampled. The parameter vectors of neural networks can be updated via randomly sampling mini-batch experience tuples *D* from replay buffer **D**, where the number of mini-batch is $|D|$.

The main critic network parameters are updated as

$$J(\theta_i) = \frac{1}{|D|} \sum_{s_t, a_t \in D} (y_t - Q_{\theta_i}(s_t, a_t))^2, i = 1,2 \tag{9}$$

where θ_i denotes the main critic network parameter vector, and the action value function Q_{θ_i} can be calculated based on the Bellman equation [15].

The target value can be expressed as

$$\begin{aligned} y_t = r_t + \gamma\Big(&\min\nolimits_{i=1,2} Q_{\theta'_i}(s_{t+1}, a_{t+1}) \\ &- \alpha \log_{\pi_\phi}(\tilde{a}_{t+1}|s_{t+1})\Big), \tilde{a}_{t+1} = \pi_\phi(\cdot|s_{t+1}) \end{aligned} \tag{10}$$

where θ'_i and ϕ denote target critic networks and actor network parameter vector, respectively. α is the temperature parameter, which determines the relative importance of entropy term versus reward, improves the randomness of the feasible policy and can be adaptively updated according to $\nabla_\alpha J(\alpha)$ [11]. $\tilde{a}_{t+1}$ emphasizes that the next action should be resampled from the policy.

The actor network parameter vector is updated according to

$$\begin{aligned} J(\phi) = \frac{1}{|D|} \sum_{s_t, a_t \in D} \big(&\min\nolimits_{i=1,2} Q_{\theta_i}(s_t, a_t) \\ &- \alpha \log \pi_\phi(a_t|s_t)\big) \end{aligned} \tag{11}$$

The target critic networks are updated according to

$$\theta'_i \leftarrow \beta\theta_i + (1-\beta)\theta'_i, i = 1,2 \tag{12}$$

where β is the soft update parameter.

B. The Equivalent Transformation of Optimization Problem

In this resource allocation problem, the index of each user is artificially prescribed, if permute indexes of users, for example, mark the *i*th user as $\sigma(i)$th, the new optimization problem is equivalent to the original. The concrete analysis is as follows.

Consider the symmetric group S_N of finite set $I_N = \{x_1, x_2, \ldots, x_N\}$, where group elements are bijection of the set I_N to itself. Assume $\sigma : I_N \to I_N$ is a permutation, the group operation is defined as the combination of the mapping, and there are total *N*! elements in S_N. The group elements S_N can be written as two-line notation [16], if $I_N = \{x_1, x_2, \ldots, x_N\}$, the two-line notation for σ is

$$\sigma = \begin{pmatrix} x_1 & x_2 & \cdots & x_N \\ \sigma_{(x_1)} & \sigma_{(x_2)} & \cdots & \sigma_{(x_N)} \end{pmatrix} \tag{13}$$

where the top row lists the elements of I_N, and the bottom row lists, under each element of I_N, its permutation under σ. Note that the two-line notation for a permutation is not unique. Given a different enumeration for I_N, both rows change accordingly.

Take transmit power **p** as an example, if $M = 2$, $N = 3$, $\mathbf{p} = \begin{bmatrix} p_{11}, & p_{12}, & p_{13} \\ p_{21}, & p_{22}, & p_{23} \end{bmatrix}$, given a case of permutation $\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$, then the power variable can be transformed into $\mathbf{p}^{(\sigma)} = \begin{bmatrix} p_{13}, & p_{11}, & p_{12} \\ p_{23}, & p_{21}, & p_{22} \end{bmatrix}$.

Therefore, the allocated transmit power in every time slot can be written as

$$\boldsymbol{p}^{(\sigma)} = \begin{bmatrix} p_{1\sigma_{(1)},t}, & \cdots & p_{1\sigma_{(N)},t} \\ \vdots & \ddots & \vdots \\ p_{M\sigma_{(N)},t}, & \cdots & p_{M\sigma_{(N)},t} \end{bmatrix}_{M \times N} \tag{14}$$

Similarly, channel state $\boldsymbol{x}$ and channel gain $\boldsymbol{g}$ should change indexes synchronously as the above permutation σ. The data rate of *j*th user can be rewritten as

$$\begin{aligned} c^{(\sigma)}_{j,t} = \sum_{i=1}^{M}\sum_{k=1}^{K}\Bigg(& x^k_{i\sigma(j),t}\tau_{i\sigma(j),t}w \\ & \log(1 + \frac{p_{i\sigma(j),t}\, g^k_{i\sigma(j),t}}{I_{i\sigma(j),t}})\Bigg) \end{aligned} \tag{15}$$

The total data rate remains unchanged since the index of each user is artificially devised, that is $C^{(\sigma)}_t = \sum_{j=1}^{N} c^{(\sigma)}_{j,t} = C_t$, and it can be deduced that the fairness index and the switching penalty are also unchanged.

In conclusion, the original optimization problem (*P*0) is equivalently transformed into

$$\begin{aligned} (P1) \quad \max_{\boldsymbol{x}^{(\sigma)}, \boldsymbol{p}^{(\sigma)}, \boldsymbol{\tau}^{(\sigma)}} \quad & \sum_{t=0}^{T} (y_1 C^{(\sigma)}_t + y_2 f^{(\sigma)}_t - y_3 h^{(\sigma)}_t) \\ s.t. \quad & \mathrm{C1}^{(\sigma)} - \mathrm{C4}^{(\sigma)}. \end{aligned} \tag{16}$$

C. The Data Augmentation Schemes for Solution

During the SAC training process, the generation of experience tuple via agent-to-environment interactions usually leads to the quite costly and which further retards the learning speed of algorithm. If (s_t, a_t, r_t, s_{t+1}) can be generated by equivalent permutation based on a symmetric group, which will effectively enrich the dataset with lower complexity, accelerate learning speed and even beneficial the accuracy of the model.

According to (13), the original policy formulation a_t can be transformed into

$$a_t^{(\sigma)} = \left\{p_{1\sigma(1),t}, \ldots, p_{M\sigma(N),t}; \tau_{1\sigma(1),t}, \ldots, \tau_{M\sigma(N),t}\right\}. \tag{17}$$

Similarly, the original state can be written

$$s_t^{(\sigma)} = \Big\{x^1_{1\sigma(1),t}, \ldots, x^K_{i\sigma(j),t}; g^1_{1\sigma(1),t}, \ldots, g^K_{i\sigma(j),t}; e_{\sigma(1),t}, \ldots, e_{\sigma(j),t}\Big\}. \tag{18}$$

Moreover, the reward function r_t is closely related to $(P1)$ and thus remains unchanged as (16), that is $r_t(s_t^{(\sigma)}, a_t^{(\sigma)}) = r_t(s_t, a_t)$. We can finally obtain the transformed experience tuple $(s_t^{(\sigma)}, a_t^{(\sigma)}, r_t, s_{t+1}^{(\sigma)})$.

Through the above subsection analysis of the symmetric group, it can be deduced that the maximum number of permutations is $N!$ in this N users scenario. Therefore, $N!$ experience tuples $(s_t^{(\sigma)}, a_t^{(\sigma)}, r_t, s_{t+1}^{(\sigma)})$ can be generated. Let S denote the set of new experience tuples, which is

$$S = \Big\{(s_t^{(1)}, a_t^{(1)}, r_t, s_{t+1}^{(1)}), \ldots (s_t^{(j)}, a_t^{(j)}, r_t, s_{t+1}^{(j)}), \ldots (s_t^{(N!)}, a_t^{(N!)}, r_t, s_{t+1}^{(N!)})\Big\} \tag{19}$$

Then, two data augmentation schemes based on equivalent permutations are designed for the SAC algorithm. The specific methods are as follows,

1) The Random Data Augmentation for Experience Replay Buffer: Randomly generate mini-batch experience tuples Λ, $\Lambda \subset S$, where the number of mini-batch is denoted as λ, and then put them into the experience replay buffer $\boldsymbol{D}$, which is updated as follows

$$\mathbf{D} \leftarrow \mathbf{D} \cup \{(s_t, a_t, r_t, s_{t+1})\} \cup \ldots \cdots \cup \{(s_t^{(\lambda)}, a_t^{(\lambda)}, r_t, s_{t+1}^{(\lambda)})\} \tag{20}$$

the random scheme will be terminated when the experience replay buffer $\boldsymbol{D}$ reaches its upper limit. Therefore, this scheme partly satisfies the demands of experience tuples diversity with comparatively low complexity in the initial stage of algorithm. This random data augmentation scheme for SAC based on permutation equivalence is termed PESAC.

2) The Adaptive Data Augmentation for Experience Replay Buffer: At the initiative of the SAC algorithm, there are X, $(X \gg N)$, experience tuples are randomly generated and then put into experience replay buffer $\mathbf{D}$. The number of augmented experience tuples λ_v in every episode can be expressed as

$$\lambda_v = \lfloor X\omega^{v-1} \rfloor \tag{21}$$

where $\omega \in (0, 1)$ is attenuation factor, v denotes episode and $v \geq 1$. This formula indicates that the number of experience tuples adaptively decreases with the training of the algorithm until the data of replay buffer has no more augmentation.

Therefore, the update of experience replay buffer $\mathbf{D}$ in every episode is given by

$$\mathbf{D} \leftarrow \mathbf{D} \cup \{(s_t, a_t, r_t, s_{t+1})\} \cup \ldots \cdots \cup \{(s_t^{(\lambda_v)}, a_t^{(\lambda_v)}, r_t, s_{t+1}^{(\lambda_v)})\} \tag{22}$$

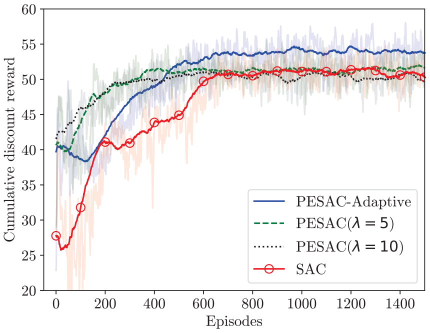


Fig. 1. The cumulative discount reward versus episode.

where $\{(s_t^{(\lambda_v)}, a_t^{(\lambda_v)}, r_t, s_{t+1}^{(\lambda_v)})\} \subset S$. This adaptive data augmentation for SAC based on permutation equivalence is named as PESAC-Adaptive.

IV. Simulation Result

In this section, the effectiveness and availability of the proposed algorithms are proved. The main simulation parameters are as follows, the maximum allocated power p^{max} is set as 28dBm, the noise o is -110dBm, and the bandwidth of channel w is 10MHz. The deep learning framework exploited in the experiments is Pytorch, all of the neural networks are four-layer fully connected networks with the learning rate 1×10^{-5}. During the training process of algorithms, the size of mini-batch $|D|$ is 256, the size of experience replay buffer is 1×10^5, the discount factor γ is 0.99, the maximum time slot per episode T is 32, and the soft update parameter β is 0.01. Moreover, the number of base stations and users is $M = 3$ and $N = 15$, respectively. The attenuation factor ω for the adaptive scheme is 0.991, and the parameter X is 128.

Fig. 1 shows the cumulative discount return of different algorithms versus episodes. The curves correspond to the mean and the shaded region to the minimum and maximum returns over the training. As can be seen that the training speed of the two proposed algorithms performs better than that of the SAC in the early training stage. The random scheme with $\lambda = 5, 10$ outperforms others in the early stage due to its increasing diversity of experience tuples, and there was no significant difference in the late stage since the augmentation is terminated and the algorithm degenerates to the original SAC. By comparison, the adaptive scheme performs better than SAC and converges fastest in the late training stage, since it meets the diverse demands of experience tuples in the early stage and the decorrelation demands in the late stage. Moreover, the downward and upward trends for algorithms mainly attribute to the random exploration of the agent.

Then, the experiments reveal the relationship between evaluation metrics and time slots after 1500 training episodes, where network parameter vectors (θ_i, θ_i' and ϕ) remain unchanged, and each metric comparison is the average of 20 times experiments due to the dynamic cellular network environment.

Fig. 2 shows the average data rate of different algorithms versus time slots. It is observed that the data rate of the

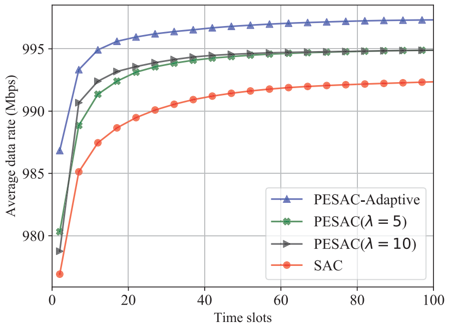
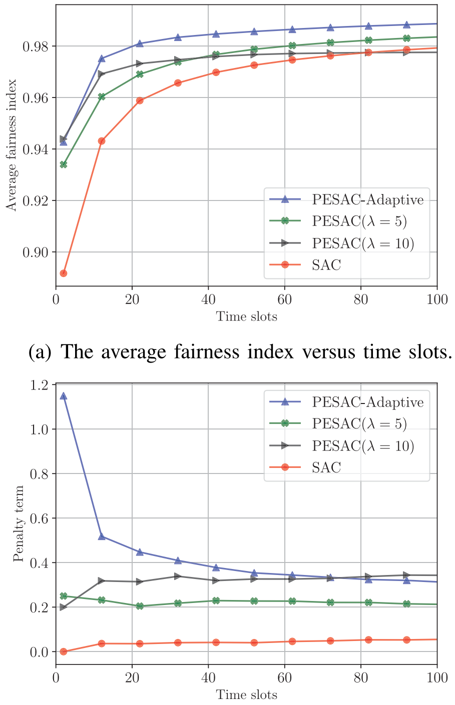


Fig. 2. The comparison of the average data rate of different algorithms versus time slots.



(a) The average fairness index versus time slots.

(b) The penalty term versus time slots.

Fig. 3. The performance of different algorithms versus time slots.

proposed PESAC-Adaptive algorithm outperforms others during time slots and there are no significant differences for PESAC with different λ, which are consistent with the final performance of Fig. 1. The experiment result shows that the adaptive scheme benefits to the training model of optimization objective.

Fig. 3(a) and (b) show the average fairness index and penalty term of different algorithms versus time slots. It can be observed that the fairness index of the PESAC-Adaptive is better than that of others. However, the penalty term of PESAC-Adaptive is worse than that of others, and the random scheme also inevitably boosts the switching penalties, this is because the augementation schemes enlarge the exploration space resulting in more frequent switching. As time slots increase, the fairness index of the different algorithms reaches a near level, which further proves the superiority of data augmentation schemes for the SAC algorithm.

V. Conclusion

This letter investigates cooperative resource allocation in SWIPT-enabled cellular networks where channel conditions are time-varying. SAC algorithm is adopted to solve the mixed integer non-linear optimization problem. Inspired by data augmentation for RL, this letter analyzes the equivalent permutation of optimization problem and proposes two data augmentation schemes for the experience replay buffer of SAC, which effectively enriches the dataset with low complexity. The cumulative discount rewards reveal that the two proposed algorithms outperform the baseline in the learning speed. The experiment results also show that the data rate and the system fairness of the proposed schemes outperform that of SAC, especially for the adaptive scheme. Meanwhile, the proposed schemes also partly increase the switching penalties due to enlarging the exploration space.

References

[1] D. Zhai, R. Zhang, J. Du, Z. Ding, and F. R. Yu, “Simultaneous wireless information and power transfer at 5G new frequencies: Channel measurement and network design,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 1, pp. 171–186, Jan. 2019.

[2] X. Wang and M. C. Gursoy, “Coverage analysis for energy-harvesting UAV-assisted mmWave cellular networks,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 12, pp. 2832–2850, Dec. 2019.

[3] M. D. Renzo and W. Lu, “System-level analysis and optimization of cellular networks with simultaneous wireless information and power transfer: Stochastic geometry modeling,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2251–2275, Mar. 2017.

[4] D. Xu and Q. Li, “Cooperative resource allocation in cognitive radio networks with wireless powered primary users,” *IEEE Wireless Commun. Lett.*, vol. 6, no. 5, pp. 658–661, Oct. 2017.

[5] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, “Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service,” *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 122–134, Dec. 2020.

[6] C. J. Chun, J. M. Kang, and I. M. Kim, “Adaptive rate and energy harvesting interval control based on reinforcement learning for SWIPT,” *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2571–2574, Dec. 2018.

[7] J.-M. Kang, “Reinforcement learning based adaptive resource allocation for wireless powered communication systems,” *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1752–1756, Aug. 2020.

[8] H. Zhang, H. Zhang, K. Long, and G. K. Karagiannidis, “Deep learning based radio resource management in NOMA networks: User association, subchannel and power allocation,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2406–2415, Oct.–Dec. 2020.

[9] J. Luo, J. Tang, D. K. So, G. Chen, K. Cumanan, and J. A. Chambers, “A deep learning-based approach to power minimization in multi-carrier NOMA with SWIPT,” *IEEE Access*, vol. 7, pp. 17450–17460, 2019.

[10] H. Peng and X. Shen, “Multi-agent reinforcement learning based resource management in MEC-and UAV-assisted vehicular networks,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[11] T. Haarnoja, A. Zhou, K. Hartikainen, and G. Tucker, “Soft actor-critic algorithms and applications,” 2018, *arXiv:1812.05905*.

[12] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[13] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, “Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 1861–1870.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[16] Á. Seress, *Permutation Group Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.