Reinforcement Learning-Based Global Programming for Energy Efficiency in Multi-Cell Interference Networks

Ramprasad Raghunath, Bile Peng, Karl-Ludwig Besser, and Eduard A. Jorswieck
Institute for Communications Technology
Technische Universität Braunschweig, Germany
Email: {r.raghunath, b.peng, k.besser, e.jorswieck}@tu-bs.de

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838408

Abstract—**With the increasing application of internet of things (IoT), the number of wirelessly transmitting devices is on a rise. It is important that the energy efficiency (EE) is maximized to reduce interference and save energy. This work explores the possibility of power control for maximum EE in wireless interference networks using reinforcement learning (RL) techniques. We apply the soft actor-critic (SAC) algorithm based on entropy regularization that allows to escape local optima and foster exploration. This enables us to solve the energy efficient power control problem with reduced complexity. We demonstrate that the obtained solutions are close to the global optimum. In contrast to supervised machine learning (ML) techniques, we do not need any kind of labeled data in the training phase. The model free approach and the unsupervised nature of RL therefore reduce the required computational effort and has a better scalability as a consequence.**

Index Terms—**Reinforcement learning, Cross entropy method, Non-convex optimization, Energy efficiency, Power allocation.**

I. Introduction

Energy efficiency (EE) in the context of wireless communication networks is defined as the ratio of the data rate to the amount of energy spent to transmit the data. It is also one of the main factors in minimizing the total cost of ownership (TCO). According to the studies by the NGMN alliance [1], the EE in wireless networks is required to increase by a factor of 2000 in the near future. EE management is one of the important factors which can contribute to sustainability of this field. The design philosophy of most wireless communication systems is to maximize capacity by using more resources. However, this may not improve the overall EE and it should be noted that these systems are mainly powered by carbon-based energy sources. It is therefore of great interest to design algorithms for a flexible resource allocation in multi-user communication systems in order to maximize EE.

Unfortunately, such EE optimization problems are non-convex and can be NP-hard in general [2]. For wireless interference networks, the global optimum for this problem can be calculated by the branch-and-bound (BB) algorithm presented in [3]. Even though this allows a computation in an offline simulation scenario, its complexity is still exponential in the number of variables and it therefore prohibits to compute the globally optimal solution. Especially in the case of a deployment with time varying channels, the EE maximization problem needs to be solved repeatedly, which requires timely updates of the power control policy.

We therefore need a different approach to flexibly determine the optimal resource allocation for interference networks. Promising new techniques for future wireless networks are machine learning (ML) algorithms [4], [5]. For the adaption to our problem, the idea is to train the system offline with a set of channel realizations, such that it learns to optimize the power allocation. For every new channel realization in the implemented system, the corresponding power allocation can easily be calculated by a simple forward-pass through the trained ML system. In [3], it is shown that feed-forward neural networks (FF-NNs) are a valid choice for calculating the optimal power control and that they are able to nearly attain the globally optimal solution. However, they belong to the class of supervised learning approaches and, therefore, require labeled training samples. Hence, we still need to compute the globally optimal solution with existing methods for a sufficient number of channel realizations. Even though, the number of required training samples can be drastically reduced [6], it is still computationally expensive and scales exponentially in the number of variables. Additionally, the training has to be carried out multiple times if there are different scenarios.

In this work, we therefore resort to less supervised ML approaches. In particular, we apply reinforcement learning (RL) in order to maximize the EE in a wireless interference network.

RL is one of the promising research areas in the ML field. An RL agent observes the environment states, makes decisions and observes its effects on the environment to improve its decision making policy in the future. Recently, RL has been an emerging tool in the areas of communications and networking due to internet of things (IoT) networks becoming more decentralized and autonomous [7]. In particular, it has been applied for distributed down-link inter-cell power control [8]. In [9], a hetrogeneous Graph neural network (hetGNN) is designed for learning to optimize power control in multi-cell networks. A deep Q-learning based model-free approach is used in [10] to compute the power allocation of transmitters in real time on a

This work is supported in part by the German Research Foundation (DFG) under grant JO 801/23-1. and the Federal Ministry of Education and Research (BMBF, Germany) as part of the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K.

mobile ad hoc network (MANET) scenario.

A distributed Q-learning algorithm is used in [11], for power control in wireless networks and also used in wireless body area networks (WBAN) [12] for dynamic power control. In [13] an indirect RL approach is used on the problem of power control in wireless communication. The authors of [13] believe there is a reduction in the number of episodes to converge to an optimal result when compared with the direct RL approach.

In contrast to the existing work, our proposed approach tries to sample decisions from a continuous space and use entropy regularization in policy learning to avoid local optima and encourage exploration. The contribution of this work is as follows:

- We propose to solve the problem of power control for maximum EE in a wireless interference network with the soft actor-critic (SAC) framework using both single agent and multi-agent reinforcement learning (MARL).
- Numerical results are put forward to check the performance of the suggested approaches. The evaluation results show that both single-agent and multi-agent RL algorithms achieve a performance close to the global optimum. The globally optimal solution is obtained by a BB algorithm [3], which has a significantly higher complexity when employed in a communication system with varying channels.

In the following Section II, we will state the considered system model and formulate the problem. The proposed solution of the EE maximization problem together with required background on the applied RL techniques is described in Section IV. A numerical evaluation with a comparison to the globally optimal solution and supervised learning algorithms is shown in Section V. Section VI concludes the paper.

II. System Model and Problem Formulation

In this work, we consider the uplink of a multi-cell interference network as described in [3]. The scenario consists of M base stations (BSs) which serve L single-antenna users. Each user j is assigned to one BS that is denoted as $a(j)$. Each BS is equipped with n_R antennas. The received signal $\boldsymbol{y}_i \in \mathbb{C}^{n_R}$ at BS i is given as

$$\boldsymbol{y}_i = \sum_{j=1}^{L} \boldsymbol{h}_{i,j} x_j + \boldsymbol{z}_i \,, \tag{1}$$

where $\boldsymbol{h}_{i,j} \in \mathbb{C}^{n_R}$ is the channel from user j to BS i, $x_j \in \mathbb{C}$ is the symbol transmitted by user j, and $\boldsymbol{z}_i$ is zero-mean circularly symmetrical complex Gaussian noise with power σ_i^2. Each user is subject to a transmit power constraint, i.e., $p_j \leq P_j$ where p_j is the average power of x_j.

Upon maximum ratio transmission beamforming, the resulting achievable rate of user j to its intended BS $a(j)$ is then given as

$$R_j = B \log\left(1 + \frac{\alpha_j p_j}{1 + \sum_{k \neq j} \beta_{j,k} p_k}\right) \tag{2}$$

with B being the communication bandwidth, $\alpha_j = \frac{\left\|\boldsymbol{h}_{a(j),j}\right\|^2}{\sigma_{a(j)}^2}$, and $\beta_{j,k} = \frac{\left|\boldsymbol{h}_{a(j),j}^{\mathrm{H}} \boldsymbol{h}_{a(j),k}\right|^2}{\sigma_{a(j)}^2 \left\|\boldsymbol{h}_{a(j),j}\right\|^2}$. From this, we can define the EE of the link between user j and BS $a(j)$ as the ratio between the achievable rate and the power consumption, i.e.,

$$\mathrm{EE}_j = \frac{B \log\left(1 + \frac{\alpha_j p_j}{1 + \sum_{k \neq j} \beta_{j,k} p_k}\right)}{\mu_j p_j + P_{c,j}}, \tag{3}$$

with the inefficiency μ_j of user j's power amplifier and the total static power consumption $P_{c,j}$ of user j.

A. Problem Statement

Throughout the rest of this work, we will consider the weighted sum energy efficiency (WSEE) as our main performance metric, which is defined as

$$\mathrm{WSEE} = \sum_{i=1}^{L} w_i \frac{\log\left(1 + \frac{\alpha_i p_i}{1 + \sum_{j \neq i} \beta_{i,j} p_j}\right)}{\mu_i p_i + P_{c,i}}, \tag{4}$$

where the non-negative weights w_i can be used to prioritize individual links.

The motivation behind the choice of WSEE is that it enables to balance the EE levels among the users with the help of non-negative weights w_i [3]. Another motivation is that WSEE is a direct generalization of the system weighted sum-rate (WSR)

$$\mathrm{WSR} = \sum_{i=1}^{L} w_i \log\left(1 + \frac{\alpha_i p_i}{1 + \sum_{j \neq i} \beta_{i,j} p_j}\right), \tag{5}$$

obtained from WSEE by setting $\mu_i = 0$ and $P_{c,i} = 1$ for $i = 1, \ldots, L$.

Based on this definition, we aim to maximize the WSEE of the described network given the power constraint of the users, which leads to the following optimization problem

$$\begin{aligned} &\max_{\boldsymbol{p}} \ \mathrm{WSEE} \\ &\text{s.t. } 0 \leq p_i \leq P_i, \quad \text{for all } i = 1, 2, \ldots, L. \end{aligned} \tag{6}$$

If we consider maximizing individual EE jointly, we encounter conflicting objectives. Since for all j, (3) is strictly decreasing in p_k for all $k \neq j$, the only possible solution would be $p_k = 0$ for $k \neq j$ as $p_k \geq 0$. This means no power allocation vector exists that simultaneously maximizes all individual EE [3]. Problem (6) can be cast into the framework of multi-objective optimization. Here, we do not have individual rate constraint, however, there is an option to include the constraints in future.

III. Overview of Soft Actor-Critic Algorithm

In this section, we explain briefly how single and multi-agent RL works under the framework of SAC. We also explain the role of entropy regularization in policy optimization.

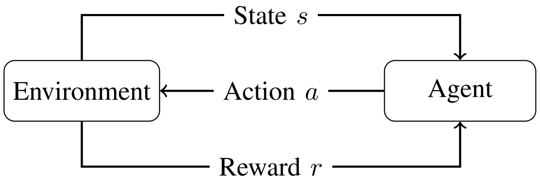

Figure 1. Framework of RL problems. The agent observes the state of the environment s and makes a decision of the action a, which determines a reward r.

A. Single Agent Reinforcement learning

SAC [14] is a state-of-the-art RL algorithm, which optimizes the behavior of an agent given a state with the trial-and-error method. As depicted in Fig. 1, the agent observes the state s of the environment, chooses an action a according to a policy π_θ parameterized by θ, i.e., $a = \pi_\theta(s)$, and obtains a reward $r = r(s, a)$. The RL problem can be formulated as

$$\begin{aligned} &\max_{\theta} \; r(s,a) \\ &\text{s.t. } a = \pi_\theta(s). \end{aligned} \tag{7}$$

SAC applies a deep neural network (DNN) as the policy to generate stochastic actions according to the state, i.e., the output of the DNN is not the action itself, but the expectation $\mu_\theta(s)$ and variance $\sigma_\theta(s)$, the actual action is sampled according to

$$a = \tanh(\mu_\theta(s) + \sigma_\theta(s) \odot \xi), \tag{8}$$

where $\xi \sim \mathcal{N}(0, 1)$ with the same dimension of the action dimension and $\odot$ denotes the scalar product. Compared to deterministic policy, which returns a deterministic action given a state, the stochastic policy fosters exploration and is more adequate for our non-convex problem with many local optima. Compared to conventional Gaussian stochastic policy, where $a = \mu_\theta(s) + \sigma \odot \xi$, (8) has the following two differences:

- The standard deviation σ is the output of the DNN rather than a constant. This enables adjustment of the exploration range.
- The $\tanh$ function restricts the action between -1 and 1, which avoids outliers deviating too much from the expectation and hence improves the training stability.

Furthermore, SAC is an entropy-regulated learning algorithm, it aims to maximize the weighted sum of reward and entropy of the action distribution. If we adopt the action distribution described in (8), the entropy is higher if σ_θ is greater. The entropy-regulation therefore encourages greater exploration and therefore it is less likely that the policy stucks in a poor local optimum. Only when an action with significant advantage is found, σ can be reduced such that the good action can be more precisely sampled in order to maximize r despite of the entropy reduction. The objective in (7) is then changed to

$$r(s,a) + \alpha H(\pi_\theta(\cdot|s)) = r(s,a) - \alpha \log(\pi_\theta(\cdot|s)), \tag{9}$$

where α is the coefficient of the entropy and $H(\pi_\theta(\cdot|s))$ is the entropy of the distribution $\pi_\theta(a|s)$. The second term of (9) encourages large deviation σ, which further encourages exploration, as described above.

After we have enough data samples of state, action and reward, the policy itself is optimized such that the Kullback–Leibler (KL) divergence between the objective (9) and the action distribution is minimized, i.e.,

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\text{KL}} \left(\pi'(\cdot|s) \middle\| \frac{\exp\left(\frac{1}{\alpha} r(s,a) - \log(\pi_\theta(\cdot|s))\right)}{Z^{\pi_{\text{old}}}(s)} \right), \tag{10}$$

where Π is the set of possible policies, $\mathrm{D}_{\text{KL}}(A\|B)$ denotes the KL divergence between distributions A and B, $Z^{\pi_{\text{old}}}(s_t)$ is the normalization factor. The intuition behind (10) is that the action distribution given s should be close to the second distribution inside the KL divergence expression. If reward r is particularly high with a certain action a, μ should be close to a and σ should shrink, otherwise σ should be kept big because $\pi_\theta(\cdot|s)$ is independent from a.

B. Multi-Agent Reinforcement Learning

Problem (7) assumes an agent that determines transmit powers of all users, which leads to high computational complexity, if we have a large number of users because the dimension of the action space is the number of users and exploration in high dimensional space is difficult (known as curse of dimensionality, see e.g., [15]). In fact, the optimal power control strategy of every user should be identical because all agents use the same policy. This fact can significantly reduce the complexity, however, is not reflected in the original RL problem. In the following, we introduce the MARL [16] as an extension of RL which addresses this problem.

We consider a decision problem of multiple agents with the same policy π_θ. Agent i observes a state s_i and makes a decision $a_i = \pi_\theta(s_i)$. The reward of agent i not only depends on its own state s_i and action a_i but also on actions of other agents a_j where $j \neq i$. Hence, the problem can be formulated as

$$\begin{aligned} &\max_{\theta} \; \sum_{i=1}^{I} r(s_i, a_1, \dots, a_I) \\ &\text{s.t. } a_i = \pi_\theta(s_i) \quad \text{for } i = 1, 2, \dots, I. \end{aligned} \tag{11}$$

where I is the total number of agents.

IV. Proposed Solutions with (Multi-Agent) Reinforcement Learning

We propose to solve problem (6), formulated in Section II, with SAC in the MARL framework, which are briefly reviewed in Section III. Note that the original RL and the SAC algorithm consider problems with time dynamics and optimize a sequence of actions along time. Since there is no time dynamics in the considered system, we assume a time horizon of one and simplify the description accordingly.

Table I
COMPLEXITIES FOR TRAINING AND EVALUATION PHASES OF THE DIFFERENT ALGORITHMS TO SOLVE THE RESOURCE ALLOCATION PROBLEM

Algorithm	Training Phase	Application Phase
Globally Optimal BB	—	High
Supervised Learning (FF-NN)	Very High	Low
RL (Proposed scheme)	High	Low

A. RL for Power Allocation

We apply the above-introduced algorithms for our EE maximization problem. In the context of multi-cell interference networks, an agent represents a power controller in the base station. However, in case of single-agent framework, the agent is responsible for power allocation of all the users. In case of the multi-agent framework, one agent is responsible for one user. In the single-agent framework, we define the state as the channel gains of all signal and interference channels following a given order, define the action as the transmit powers of all BSs and the reward as the WSEE.

In the multi-agent framework, we define the state as the channel gains of all signal and interference channels following a given order and i (index of the power controller in BS), define the action as the transmit power allocated by power controller i in BS and the reward as the WSEE. In this way, one power controller in each BS is controlled by one agent and the action space dimension is reduced from L to 1. The reward is the same for all agents such that altruistic cooperation is encouraged. All agents share the same policy according to which actions are selected and different actions/power allocations are made according to different states/channel gains.

One of the major advantages of using RL over FF-NN is captured in Table I. The training effort for FF-NN is extremely high, because we need to generate a sufficient amount of training labels (solving the globally optimal solution a lot of times) on top of the actual training. Although the training effort for both RL and FF-NN is similar, the effort required to create a dataset for the latter is very high because a dataset for supervised learning contains global optimum power allocation and this increases computational complexity. In contrast, the unsupervised learning nature of RL accounts for its lower computational efforts in its training phase. Considering the application phase, both approaches have similar complexity because only the channel gains without labeled data are needed. Besides, the high sample efficiency as a result of the off-policy property of the SAC algorithm further reduces the training complexity. Comparing single-agent and multi-agent frameworks, the multi-agent RL utilizes the fact that all agents (BSs) are symmetric by applying the same policy to all agents, therefore significantly reduces the neural network (NN) complexity. On the contrary, transmission powers of different BSs in the single-agent RL framework are different dimensions of the action.

Table II
HYPERPARAMETERS USED TO TUNE THE MODELS FOR FOUR USER SCENARIO

Parameters	Single Agent	Multi Agent
Learning rate	10^{-5}	10^{-5}
Framework	PyTorch	PyTorch
Batch size	2048	2048
Soft update coefficient τ	0.001	0.001
Entropy coefficient	*auto*	*auto*
Iterations	600 000	700 000

V. NUMERICAL EVALUATION

In this section, we numerically demonstrate the effectiveness of our proposed RL implementation to solve (6). In order to being able to compare our results with the ones from [3], we use the same parameters for the communication system, which we will restate from [3, Sec. VI] in the following. There are $L = 4$ users in a square area with an edge length of $2\,\mathrm{km}$. They are served by $M = 4$ BSs, which are placed at the coordinates $(0.5, 0.5)\mathrm{km}$, $(1.5, 0.5)\mathrm{km}$, $(0.5, 1.5)\mathrm{km}$, and $(1.5, 1.5)\mathrm{km}$. Each BS is equipped with $n_R = 2$ antennas. The model of the path-loss follows [17], where we set the carrier frequency to $1.8\,\mathrm{GHz}$ and the power decay factor to 4.5. Fast fading is modeled as circularly symmetric complex Gaussian with zero mean, unit variance. The static power consumption and power amplifier inefficiency are equal to $P_{c,i} = 1\,\mathrm{W}$ and $\mu_i = 4$, $i = 1, \dots, L$, respectively. The noise power at each receiver is given as $\sigma^2 = F\mathcal{N}_0B$, where $F = 3\,\mathrm{dB}$ is the receiver noise figure, $B = 180\,\mathrm{kHz}$ is the communication bandwidth, and $\mathcal{N}_0 = -174\,\mathrm{dBm/Hz}$ is the noise spectral density. All users have the same maximum transmit powers $P_1 = \cdots = P_L = P_{\max}$.

Using the channel information generated with the parameters mentioned above, we model our problem as markov decision process (MDP) in the framework of OpenAI's Gym environment. After each step, EE is returned corresponding to the predicted actions from the current policy of the agent. Along with this, the current state of the environment, status of the episode and some additional information to debug its working is also returned.

With the available transmit power, the step reward is calculated as the EE of the network and fed back to the agent.

An existing implementation of the proposed SAC algorithm from Stable Baselines 3 [18] and Ray RLlib [19] have been used for single and multi-agent RL, respectively. Both [18] and [19] use DNN with 2 fully-connected layers of 256 neurons each as actor and critic. These layers are activated by the rectified linear unit (ReLU) function which is non-linear in nature. Batches of 2048 samples from replay buffer are passed as inputs to the DNN.

The hyper-parameters used to initialize the agents are consolidated in Table II. These parameters were empirically determined after several iterations.

In order to allow for a fair comparison with the results from [3], we use the same data set [20]. The training data is

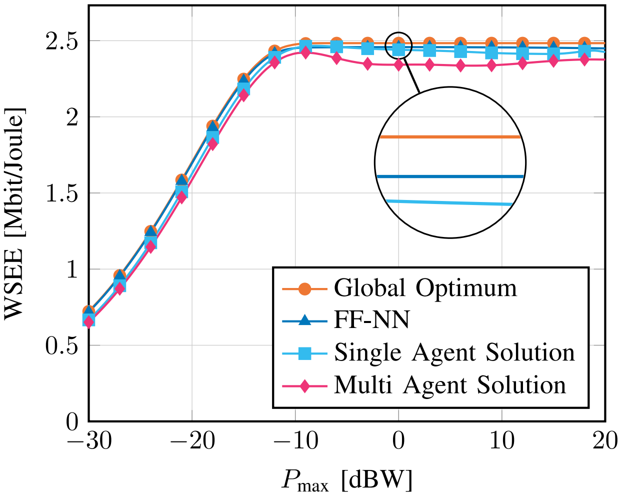

Figure 2. WSEE performance of the proposed single and multi agent RL algorithms compared with the globally optimal WSEE and the FF-NN approximation from [3] for the scenario with four users.

generated from 2000 independent and identically distributed (i.i.d.) channel realizations. The users are assigned to the access point with the strongest effective channel gain α_i. These channel gains in combination with maximum allowed power $P_{\max} = -30, \dots, 20\,\text{dBW}$ in $1\,\text{dBW}$ steps give us a training set of 102 000 samples. For testing the trained models, we used 510 000 samples generated from 10 000 channel realizations and averaged the performance of the model over these samples. It should be emphasized that the test data is not a part of the environment during the training process.

The full data set and the source code to reproduce the results are publicly available at [20] and [21], respectively.

The performance of the single agent and multi agent systems is shown in Fig. 2. For comparison, we additionally show the global optimum derived by the BB algorithm from [3] and the WSEE results of the supervised FF-NN from [3, Sec. VI]. As mentioned above, the same data set [20] is used to generate the results for all compared methods. The first interesting observation is that the proposed RL approach can perform close to the global optimum and the supervised learning approach *without* the need of labeled training data. Even though the suggested approach achieves a slightly lower WSEE, it maintains stability at lower powers at a significantly reduced complexity, cf. Table I. At higher powers, both single and multi agent show signs of deviation from the global optimum. Specifically up to around $P_{\max} = -10\,\text{dBW}$, the RL approaches perform comparable to FF-NN and global optimum.

It is also interesting to note that the single agent model performs slightly better than multi agent model. This behavior in multi-agent approach is due to an increased size of joint action space. As the number of agents increase, there is a growth in the joint action space. This is known as curse of dimensionality in MARL. The performance difference is also because each agent affects the environment based on its action. This results in a non-stationary environment [22].

A. Extension to an Increased Number of Users

The results discussed above are for a scenario with $L = 4$ users. In order to demonstrate the scalability of the suggested approach, we now increase the number of users to $L = 7$ and $n_r = 4$ antennas per BS.

The training data has been generated from 6000 i.i.d. channel realizations and 1000 channels have been used for test data. This gives a total of 306 000 and 51 000 samples for training and testing respectively [20].

Since the problem is invariant under permutation of the users, we can increase the size of the training data during the course of training just by permuting the rows and columns of channel matrix. After each environment step, in the reset method, a new random permuted data is used as state. This enables the agents to learn about the invariance against permutation of users and also generate enough data for training.

The performance of both single and multi agent approaches on the test data for 7 user scenario has been presented in Fig. 3. Again, we compare the RL approaches to the global optimum and the results obtained by supervised learning (FF-NN) from [3]. It can be seen from Fig. 3 that both single and multi-agent frameworks perform well until $-10\,\text{dB W}$ and at higher powers drop in performance. At higher powers, our suggested approach seems to be getting stuck at local optimum. The hyper-parameters used to tune the model for seven user scenario are consolidated in Table III. The results were achieved by decreasing the learning rate to 10^{-6} as compared to four user scenario because it was performing worse with the same parameters as before and the learning rate represent how fast the gradients are updated. The batch size is also increased because overall dimension of the problem is increased. Also the soft update coefficient was set to 1.15×10^{-4}. This parameter in particular is very sensitive to tuning. Here, future work is required to improve the performance and stability of training.

Table III
HYPER-PARAMETERS FOR SEVEN USERS SCENARIO

Parameters	Single Agent	Multi Agent
Learning rate	10^{-5}	10^{-6}
Batch size	32 768	32 768
Soft update coefficient τ	10^{-4}	1.15×10^{-4}
Entropy coefficient	*auto*	*auto*
Iterations	2m	1.3m

VI. Conclusion

The EE is of vital importance in the future wireless communication systems. Its non-convex nature makes it very difficult to find the globally optimal power allocation to maximize the EE. In our previous works, the BB algorithm has been successfully applied to find the global optimum. However, its high complexity makes it difficult for real-time applications. In this paper, we apply the model-free RL algorithm, which optimizes the power control strategy with the trial-and-error method and applies the entropy regularization to encourage

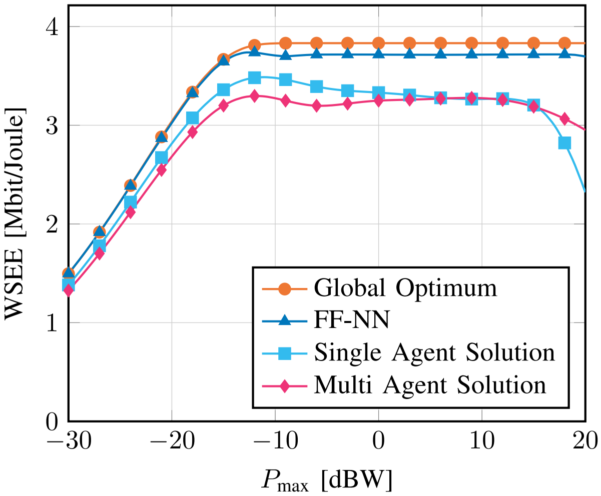

Figure 3. WSEE performance of the proposed single and multi agent RL algorithms compared with the globally optimal WSEE and the FF-NN approximation from [3] for the scenario with seven users.

exploration, such that the solution does not stuck in a poor local optimum. Evaluation results show that the proposed solution achieves a performance close to the global optimum found by the BB algorithm. The main advantage of the proposed algorithm is its low complexity in application. In addition, in the multi-agent framework, different power controllers in BSs are represented by different agents with the same policy. In this way, the symmetry of BSs is utilized and the solution is more scalable compared to the single-agent solution.

This work can be further expanded such that an environment is created to include a system model which can simulate the connection between certain number of users and BSs to transmit random packets of information so that we have large amount of dataset and can also take advantage of a feature in RL called continuous learning. This allows the RL model to be implemented at hardware level and BS can be smart. The current models can also be further improved by fine-tuning the hyper-parameters.

Another optimization method called cross entropy method (CEM), a tool for solving estimation and optimization problems, based on Kullback–Leibler (or cross-entropy) minimization [23], can be applied to this problem of optimizing the power controller for maximum EE. This method operates similar to SAC, by encouraging exploration initially.

References

[1] "NGMN 5G white paper," NGMN Alliance, Feb. 2015.

[2] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[3] B. Matthiesen, A. Zappone, K.-L. Besser, E. A. Jorswieck, and M. Debbah, "A globally optimal energy-efficient power control framework and its efficient implementation in wireless interference networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3887–3902, 2020. arXiv: 1812.06920 [cs.IT].

[4] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137 184–137 206, 2019.

[5] F.-L. Luo, Ed., *Machine Learning for Future Wireless Communications*. Wiley-IEEE Press, 2020.

[6] K.-L. Besser, B. Matthiesen, A. Zappone, and E. A. Jorswieck, "Deep learning based resource allocation: How much training data is needed?" In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Atlanta, GA, USA: IEEE, May 2020.

[7] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[8] E. Ghadimi, F. Davide Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.

[9] J. Guo and C. Yang, "Learning power control for cellular systems with heterogeneous graph neural network," *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2021.

[10] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[11] A. Ornatelli, A. Tortorelli, and F. Liberati, "A distributed reinforcement learning approach for power control in wireless networks," in *2021 IEEE World AI IoT Congress (AIIoT)*, IEEE, May 2021.

[12] R. Kazemi, R. Vesilo, E. Dutkiewicz, and R. Liu, "Dynamic power control in wireless body area networks using reinforcement learning with approximation," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, Sep. 2011.

[13] A. Udenze and K. McDonald-Maier, "Indirect reinforcement learning for autonomous power configuration and control in wireless networks," in *2009 NASA/ESA Conference on Adaptive Hardware and Systems*, IEEE, Jul. 2009.

[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 *International Conference on Machine Learning*, PMLR, 2018, pp. 1861–1870.

[15] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[16] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[17] G. Calcev, D. Chizhik, B. Goransson, S. Howard, H. Huang, A. Kogiantis, A. F. Molisch, A. L. Moustakas, D. Reed, and H. Xu, "A wideband spatial channel model for system-wide simulations," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 389–403, Mar. 2007.

[18] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann. (2019). "Stable baselines3," [Online]. Available: https://github.com/DLR-RM/stable-baselines3.

[19] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "RLlib: Abstractions for distributed reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2018.

[20] B. Matthiesen, A. Zappone, K.-L. Besser, E. Jorswieck, and M. Debbah, *Accompanying data to paper "A globally optimal energy-efficient power control framework and its efficient implementation in wireless interference networks"*, IEEE DataPort, Oct. 2020.

[21] R. Raghunath, B. Peng, K.-L. Besser, and E. Jorswieck. (2021). "Reinforcement learning-based global programming for energy efficiency in multi-cell interference networks, Source code," [Online]. Available: https://gitlab.com/ichbinram/rl-ee-opt.

[22] L. Busoniu, R. Babuska, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[23] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L'Ecuyer, "The cross-entropy method for optimization," in *Handbook of Statistics – Machine Learning: Theory and Applications*, ser. Handbook of Statistics, C. Rao and V. Govindaraju, Eds., vol. 31, Elsevier, 2013, ch. 3, pp. 35–59.