A Deep Q-Learning Bisection Approach for Power Allocation in Downlink NOMA Systems

Marie-Josepha Youssef, Charbel Abdel Nour, Xavier Lagrange, Catherine Douillard

A Deep Q-Learning Bisection Approach for Power Allocation in Downlink NOMA Systems

Marie-Josepha Youssef, Charbel Abdel Nour, Xavier Lagrange and Catherine Douillard

***Abstract*—In this work, we study the weighted sum-rate maximization problem for a downlink non-orthogonal multiple access (NOMA) system. With power and data-rate constraints, this problem is generally non-convex. Therefore, a novel solution based on the deep reinforcement learning (DRL) framework is proposed for the power allocation problem. While previous work based on DRL restrict the solution to a limited set of possible power levels, the proposed DRL framework is specifically designed to find a solution with a much larger granularity, emulating a continuous power allocation. Simulation results show that the proposed power allocation method outperforms two baseline algorithms. Moreover, it achieves almost 85% of the weighted sum-rate obtained by a far more complex genetic algorithm that approaches exhaustive search in terms of performance.**

***Index Terms*—Non-orthogonal multiple access, deep reinforcement learning, weighted sum-rate maximization, successive interference cancellation stability.**

I. Introduction

Future communication networks are expected to support a myriad of new applications with a diverse set of requirements [1]. In addition to providing higher data rates, future cellular systems must account for a massive number of connected devices with different priority levels in terms of rate, latency and reliability needs. Among the identified promising technologies to support the new mobile traffic needs, non-orthogonal multiple access (NOMA) holds a key position [2], [3]. Indeed, NOMA is able to schedule multiple users on the same time/frequency channel, increasing both the number of active devices and system spectral efficiency.

To schedule multiple users on the same time/frequency channel, NOMA exploits power domain user multiplexing. Hence, the application of efficient power allocation procedures becomes of utmost importance. Therefore in recent literature, significant attention has been devoted to power allocation in NOMA systems to achieve different objectives such as to maximize fairness, rate, or energy efficiency [3].

To account for different user priority levels in terms of data rate, reliability and latency requirements, the system objective can be formulated as a weighted sum-rate (WSR) function. However, the WSR maximization problem in a NOMA system is non-convex [4]. In [3], the power allocation maximizing the WSR was found for a two-user system. The authors of [4] considered a K-user system and derived the conditions on the weight values, under which the WSR problem is convex. In [5], monotonic optimization was used to derive both the subband and power allocation maximizing the WSR. However, the solution of [5] suffers from a complexity that grows exponentially with both the number of subbands and users.

Recently, the use of reinforcement learning (RL) in wireless communication systems has received significant attention since it allows to reach near-optimal solutions for non-convex problems. The RL framework is a sequential decision making method where an agent interacts with an environment with the aim of maximizing its long-term discounted reward [6]. With the large expansion in network scale, deep RL (DRL) is used to increase the efficiency of traditional RL. In [7], a resource allocation solution based on DRL was proposed to maximize the rate in an uplink NOMA system. Aiming to maximize the WSR, another power allocation method based on DRL was proposed in [8]. In [9], the authors studied a massive access system and proposed a resource allocation method based on DRL to satisfy the different quality-of-service (QoS) requirements of the users. In [10], a solution based on DRL was proposed for a downlink NOMA system, where the base station (BS) is confronted with the three choices of maintaining, increasing or decreasing by a constant value the power level at each timeslot. It should be noted that the power allocation problem involves continuous variables while DRL inherently deals with discrete-type action and state spaces. Hence, almost all previous work that studied power allocation in the context of DRL proceeded to use a finite number of discrete power levels [7]–[9].

In this work, we study the WSR optimization problem in a downlink NOMA system and introduce a new DRL-based power allocation technique. Contrary to previous work, the proposed technique starts from L discrete power levels and progressively refines the solution. It does so by varying the bounds of the search space based on the chosen power levels and achieved performance. To the best of our knowledge, this is the first work that aims at finding a quasi-continuous solution, i.e., having a much larger granularity, for the power allocation problem, based on DRL.

The rest of this paper is organized as follows. The system model is presented in section II. In section III, an overview of the DRL framework along with the proposed power allocation method are detailed. Simulation results are provided in section IV and conclusions in section V.

II. System Model

Consider the downlink of a communication system consisting of one single-antenna BS and K single-antenna users uniformly deployed over the cell. Let P_{max} denote the BS power budget. Accounting for both small scale Rayleigh fading and large scale fading (i.e., path-loss and log-normal shadowing), the channel gain between user k and the BS at timeslot t is

M. J. Youssef, C. Abdel Nour and C. Douillard are with IMT Atlantique, LabSTICC, UMR CNRS 6285, F-29238 Brest, France, (e-mail: marie-josepha.youssef@imt-atlantique.fr; charbel.abdelnour@imt-atlantique.fr; catherine.douillard@imt-atlantique.fr). X. Lagrange is with IMT Atlantique, IRISA, UMR CNRS 6074, F-35700 Rennes, France (e-mail: xavier.lagrange@imt-atlantique.fr).

denoted by $h_k^{(t)}$. Without loss of generality, we assume that the users are indexed in the increasing order of their channel gains; i.e., the index of user k precedes that of user k' if $h_k^{(t)} < h_{k'}^{(t)}$. The BS leverages NOMA to serve all K users simultaneously on a frequency channel of bandwidth B. The use of NOMA leads to co-channel interference between the collocated users. Therefore, signal separation at the receiver side is done using successive interference cancellation (SIC) decoding [11] in the increasing order of channel gains. Having applied SIC, the rate of user k can be expressed as:

$$R_k^{(t)} = B \log_2 \left(1 + \frac{p_k^{(t)} \left(h_k^{(t)} \right)^2}{\sum\limits_{k'=k+1}^{K} p_{k'}^{(t)} \left(h_k^{(t)} \right)^2 + N_0 B} \right), \quad (1)$$

where $p_k^{(t)}$ is the power allocated to user k at timeslot t and N_0 is the noise power spectral density. The first term in the denominator reflects the interference experienced by user k from users having a higher channel gain than k at timeslot t, i.e., those whose interference cannot be canceled using SIC.

The objective of this work is to optimize the power allocation with the aim of maximizing the WSR under minimum rate requirements. The optimization problem is formulated as:

$$\max_{\boldsymbol{p}} \quad \sum_{k=1}^{K} w_k R_k^{(t)} \quad (2)$$

$$\text{such that} \sum_{k \in \mathcal{K}} p_k^{(t)} \leq P_{max}, \quad (2a)$$

$$R_k^{(t)} \geq R_{k,min}, \forall k \in \mathcal{K}, \quad (2b)$$

$$p_k^{(t)} > \sum_{k'=k+1}^{K} p_{k'}^{(t)} \ \forall k \in \mathcal{K}, \quad (2c)$$

$$p_k^{(t)} \geq 0 \ \forall k \in \mathcal{K}, \quad (2d)$$

where w_k is the weight relative to user k. Constraint (2a) is the BS power budget constraint while (2b) is the minimum rate constraint per user. To guarantee SIC stability [2], [3], i.e., successful decoding at the user side, the user with a lower channel gain must be allocated a power that exceeds the sum power allocated to users having a higher channel gain, as expressed in constraint (2c). Indeed, as shown in [12], the power of the weaker user must be strictly larger than the sum power of the stronger users since in the opposite case, the outage probabilities of all users will be always one.

Unless the user weights satisfy the set of conditions derived in [4], optimization problem (2) remains non-convex. Hence, it cannot be solved with standard optimization techniques.

III. DRL-Based Power Allocation

In the following, a power allocation method based on the DRL framework is introduced to solve problem (2), even when the latter is non-convex. Since the power optimization problem involves continuous variables, most previous work [7]–[9] leveraging DRL use discrete power levels taken between 0 and P_{max}. That said, system performance is largely dependent on the discretization level. To avoid a prohibitive system complexity, a tradeoff is generally made where this level is significantly limited. In this work, we introduce a technique that aims at finding a quasi-continuous solution for the power allocation while keeping system complexity manageable. Next, the power allocation problem is formulated as a Markov decision process (MDP) before briefly describing the basics of DRL. Finally, a DRL-based power allocation algorithm is proposed to maximize the WSR in a NOMA system.

A. Formulating the Power Allocation Problem as a MDP

An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action set, $\mathcal{P}$ is the transition probability, i.e., $\mathcal{P}_{ss'}^{a} = \Pr(s^{(t+1)} = s' | s^{(t)} = s, a^{(t)} = a)$ is the probability of moving from a current state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ after taking action $a \in \mathcal{A}$, and r denotes the immediate achieved reward. Based on problem (2), the action space, state space, and reward function are formulated as:

- **Action space:** The action space consists of the available power levels at the BS level for the different users. We denote by l_k and u_k the lower and upper bounds of the power value search space of user k. The action space related to the power allocation process for user k is uniformly partitioned between $l_k P_{max}$ and $u_k P_{max}$, i.e., $\mathcal{A}_k = \{l_k P_{max}, (l_k + \frac{u_k - l_k}{L-1}) P_{max}, \ldots, u_k P_{max}\}$, where L is the number of available power levels. The action space of the BS regarding the allocated power values to all K users is given by: $\mathcal{A} = \prod\limits_{k \in \mathcal{K}} \mathcal{A}_k$, hence $|\mathcal{A}| = L^K$. The transmit power allocated to each user k is given by:
$$p_k = \frac{v_k}{\sum\limits_{k' \in \mathcal{K}} v_{k'}} P_{max}, \quad (3)$$
where v_k is the chosen power level for user k.
- **State space:** At each timeslot t, the network state is defined as:
$$s^{(t)} = \left\{ a^{(t-1)}, \{h_k, w_k, \xi_k, \delta_k, l_k, u_k\}, \forall k \in \mathcal{K}, r^{(t)} \right\}, \quad (4)$$
where $a^{(t-1)}$ and $r^{(t)}$ are the chosen action and the achieved reward at the previous timeslot, respectively. For each user k, $\xi_k = 1$ if its rate requirement and SIC stability constraints are satisfied. In the opposite case, $\xi_k = 0$. Variable δ_k is the number of users for whom the satisfaction of constraints (2b) and (2c) is penalized by the transmission of user k.
- **Reward function:** The reward function, designed to optimize the network objective, is chosen to be equal to:
$$r^{(t+1)} = \sum_{k \in \mathcal{K}} r_k^{(t+1)} = \sum_{k \in \mathcal{K}} \left(\xi_k w_k R_k^{(t)} - \phi \delta_k \right), \quad (5)$$
where $r_k^{(t+1)}$ is the reward of user k and ϕ is a large positive constant equal to the the sanction inflicted upon user k for penalizing some other user from reaching its requirement. The value of ϕ is set such that $r_k^{(t+1)} < 0$ if $\delta_k > 0$.

B. Overview of DRL

A RL agent learns its best strategy from observing the rewards of trial-and-error interactions with the environment. At

each timeslot t, the agent observes the state of the environment $s^{(t)}$ and takes action $a^{(t)}$ according to a possible policy π, where $\pi(s,a)$ is the probability of choosing action a under state s. Having taken action $a^{(t)}$, the environment transitions to a new state $s^{(t+1)}$ and the agent receives a reward $r^{(t+1)}$. This interaction with the environment forms an experience expressed as: $e^{(t+1)} = \{s^{(t)}, a^{(t)}, r^{(t+1)}, s^{(t+1)}\}$.

The Q-learning algorithm aims to compute the optimal strategy π maximizing the expected reward function given by:

$$Re^{(t)} = \sum_{\tau=0}^{\infty} \gamma^{\tau} r^{(t+\tau+1)}, \tag{6}$$

where $\gamma \in (0,1]$ is the discount factor for future rewards. The Q-function associated with strategy π is defined as the expected reward achieved when taking action a under state s:

$$Q^{\pi}(s,a) = \mathbb{E}\left[Re^{(t)} | s^{(t)} = a, a^{(t)} = a, \pi\right]. \tag{7}$$

The Q-learning algorithm now aims to find the optimal strategy $\pi*$ that maximizes the Q-function. Moreover, the optimal Q-function values obey the Bellman optimality condition [6]:

$$Q^{*}(s,a) = \mathbb{E}_{s'}\left[r' + \gamma \max_{a'} Q^{*}(s',a') | s^{(t)} = s, a^{(t)} = a\right], \tag{8}$$

where s' and r' are respectively the new network state and the reward achieved when taking action a in state s.

The classical Q-learning algorithm uses a lookup-table to store and update the Q-function values. With the increase of the network scale, the classical Q-learning algorithm becomes inefficient due to the required large storage capacity of the Q-table and to a long convergence time. In such cases, a deep Q-network (DQN) with experience replay [13] is used where a neural network is employed to approximate the Q-function values. The DQN network can be expressed as $Q(s,a,\theta)$, where θ represents network parameters. The task of finding the best Q-values is equivalent to searching for the best values for θ. As in classical Q-learning, the agent collects experiences from its interactions with the environment and forms a data set D. As implied by the *quasi-static target network* method [13], two DQNs are formed: the local DQN with parameters $\theta_{local}^{(t)}$ at timeslot t, and the target DQN with parameters $\theta_{target}^{(t)}$. Every T timeslots, $\theta_{target}^{(t)}$ is updated to be equal to $\theta_{local}^{(t)}$. Periodically, the agent selects a mini-batch of experiences from its memory, $D^{(t)}$, and uses it to optimize the model parameters $\theta_{local}^{(t)}$ with the aim of minimizing the loss function defined by:

$$Loss(\theta_{local}^{(t)}) = \sum_{D^{(t)}} \left(y_{DQN}^{(t)}(r',s') - Q(s,a,\theta_{local}^{(t)})\right)^2, \tag{9}$$

where $y_{DQN}^{t}(r',s') = r' + \gamma \max_{a'} Q(s',a',\theta_{target}^{(t)})$.

C. Proposed DRL-Based Power Allocation Algorithm

Similar to [14], the DQN is trained in an offline manner for N_{ep} total episodes as shown in Algorithm 1.

First, several training parameters are initialized: the discount factor γ, the initial ϵ-greedy probability $\epsilon^{(0)}$, its minimum value ϵ_{min} as well as its decay rate λ, the batch size L_b, the memory size L_M, constants ϕ and θ, the copy frequency of the target DQN weights T, and the weights of both the local and the target DQNs, θ_{local} and $\theta_{target}^{(0)}$ such that $\theta_{target}^{(0)} = \theta_{local}^{(0)}$.

In the training phase of the algorithm, at each timelot t, the agent, i.e., the BS, inputs the network state into the local DQN to obtain the Q-values of all available actions. The selected action at timeslot t is then determined by the adaptive ϵ-greedy algorithm. In other words, with probability $\epsilon^{(t)}$, a random action is selected while the action with the maximum Q-value is chosen with probability $(1-\epsilon^{(t)})$. The exploration probability is updated according to:

$$\epsilon^{(t)} = \min(\epsilon_{min}, \epsilon^{(0)}(1-\lambda)^t). \tag{10}$$

Note that the action selection is equivalent to the chosen power level for each user, i.e., $a^{(t)} = \{v_k^{(t)}, \forall k \in \mathcal{K}\}$. Having selected its action, the agent then receives a reward $r_k^{(t+1)}$, $\forall k \in \mathcal{K}$ from the environment. If $r_k^{(t+1)} \geq 0$, i.e., if $\delta_k = 0$, the lower bound of the search space for user k is increased according to:

$$l_k = (l_k + \frac{u_k - l_k}{L-1}). \tag{11}$$

If $r_k^{(t+1)} < 0$, i.e., if $\delta_k > 0$, either the power allocated to user k must be reduced, or the power allocated to users penalized by user k must be increased. Hence, the upper bound of each user $k \in \mathcal{K}$ is reduced with probability p according to:

$$u_k = \left(l_k + \frac{(L-2)(u_k - l_k)}{L-1}\right). \tag{12}$$

The purpose of this update is to allow the BS to refine the allocated power values for each user in order to improve performance while keeping a constant system complexity. In fact, by allowing for changing search spaces, starting from discrete available power values, a quasi-continuous solution for these values is achievable. In the case where $r_k^{(t+1)} < 0$, the search space for user k is reset, i.e., $l_k(t) = 0, u_k(t) = 1$, with a probability $(1-p)$. In this case, the negative obtained reward indicates that the BS made a bad decision at some timeslot. Hence, resetting the search spaces allows the BS to try different actions in the hope of finding the optimal solution maximizing the achieved reward.

The environment transitions to a new state $s^{(t+1)}$. The agent then stores the new transition in its memory as stated in step 14 of Algorithm 1.

To train the local DQN, the experience replay method [13] is employed, where mini-batch training with a batch size of L_b is adopted. With the selected mini-batch, the local DQN is trained using the RMSprop algorithm [15] to minimize the loss given in Eq. (9). Every T training steps, the weights of the target DQN are updated to be equal to the local DQN weights.

Finally, if the difference between the lower and upper bounds of all users becomes lower than some small value v, the system converges, terminating the current training episode.

IV. Simulation Results

The number of layers and the number of neurons in each layer control the ability of the DQN to approximate complex functions. When the number of layers and neurons increase,

Algorithm 1 DRL-Based Power Allocation Algorithm

Initialization:
Initialize the training parameters $\gamma, \eta, \epsilon^{(0)}, \epsilon_{min}, \lambda, T, L_b, \alpha, \theta$.
Initialize the replay memory of size L_M, the weights of the local DQN $\theta_{local}^{(0)}$ and the weights of the target DQN such that $\theta_{target}^{(0)} = \theta_{local}^{(0)}$.
Training phase:
for $n_{ep} = 0, \ldots, N_{ep}$ **do**
  for $t = 0, 1, 2, \ldots$ **do**
    Input state $s^{(t)}$ into the local DQN and obtain the Q-values relative to all actions.
    Select action $a^{(t)}$ according to the ϵ-greedy algorithm and receive reward $r_k^{(t+1)}, \ \forall k \in \mathcal{K}$.
    if $r_k^{(t+1)} \geq 0$ **then**
      Increase the lower bound of the search space of user k.
    else
      With probability p, decrease the upper bound of the search space of user k.
      With probability $(1-p)$, reset the search spaces of user k.
    end if
    The environment transitions into a new state $s^{(t+1)}$.
    Store transition $\left(s^{(t)}, a^{(t)}, r^{(t+1)}, s^{(t+1)}\right)$ in the replay memory.
    Sample a random mini-batch of size L_b of transitions from the replay memory.
    Train the local DQN with the RMSprop algorithm.
    if $t \% T = 0$ **then**
      Update the weights of the target DQN such that $\theta_{target}^{(t)} = \theta_{local}^{(t)}$.
    end if
    if $\left(u_k^{(t)} - l_k^{(t)}\right) \leq \upsilon, \ \forall k \in \mathcal{K}$ **then**
      break
    end if
  end for
end for

the functions the DQN can approximate become more complex, at the expense of an additional computational complexity. Hence, the number of layers and neurons should be chosen to strike a tradeoff between performance and computational complexity. The adopted DQN in this work consists of four hidden layers: three fully-connected (FC) layers and a dueling layer [7]. The first FC layer consists of 200 neurons while each remaining layer consists of 100 neurons. The size of the replay memory is $L_M = 10000$. The learning rate, discount factor, batch size are $\eta = 10^{-4}, \gamma = 0.9, N_b = 64$, while ϵ decreases from 0.9 to 0.01 with a decay rate $\lambda = 1 - 10^{-4}$. The target network copies the local network weights every $T = 10$ training steps. Unless otherwise stated, the number of available power levels for each user is $L = 4$. More than 10^6 steps of training are performed while the testing of the DQN performance is averaged over 5×10^3 independent experiment runs.

We consider a single cell having a radius $R_d = 500$m with one BS located at the cell center and K uniformly deployed users. Signals undergo frequency-selective Rayleigh fading with a root mean square delay spread of 500 ns, a distance-dependent path loss with a decay factor of 3.76, and a zero-mean log-normal shadowing with an 8 dB variance. The main simulation parameters are summarized in Table I.

The proposed power allocation method, denoted by 'DRL-VB', is compared with several other methods:

- A simplified version of the DRL-based method that does not adapt the search space of the power values, denoted by 'DRL-FB'. In other words, the action space of DRL-FB is fixed and given by: $\mathcal{A}_{DRL-FB} = \{0, \frac{P_{max}}{L-1}, \ldots, P_{max}\}$.
- A method based on the fractional transmit power allocation (FTPA) method [2], denoted by 'FTPA', where the power allocated to user k is given by:

$$p_k = P_{max} \frac{\left(\left(h_k^{(t)}\right)^2 / (N_0 B)\right)^{-\alpha}}{\sum_{k' \in \mathcal{K}} \left(\left(h_{k'}^{(t)}\right)^2 / (N_0 B)\right)^{-\alpha}}, \qquad (13)$$

  with $0 < \alpha < 1$ defined as the FTPA factor.
- A fixed power allocation (FPA) method where the power allocated to user k is equal to $p_k = \beta_k P_{max}$, with β_k being the FPA factor relative to user k. Note that the values of β, listed in Table I, are chosen empirically in such a way to satisfy SIC stability and optimize system performance.
- A power allocation method based on a genetic algorithm (GA) [16] that outputs a near optimal solution [17]. This method is denoted by 'GA'.

TABLE I: Simulation parameters

Transmission Bandwidth	1.25 MHz
Number of users	2, 3, 4, 5
N_0	4×10^{-18} mW/Hz
$R_{k,min}, \forall k \in \mathcal{K}$	1.5 Mbps
Cell Power Budget	5 W (37 dBm)
Number of power levels L	4
ϕ, θ	$30, 10^{-2}$
FTPA parameter α	0.5
FPA parameter β	$\{0.7, 0.3\}$ for $K = 2$, $\{0.6, 0.25, 0.15\}$ for $K = 3$, $\{0.6, 0.25, 0.1, 0.05\}$ for $K = 4$, $\{0.51, 0.3, 0.1, 0.06, 0.03\}$ for $K = 5$
Population size of GA	1000
Maximum number of iterations of GA	3000

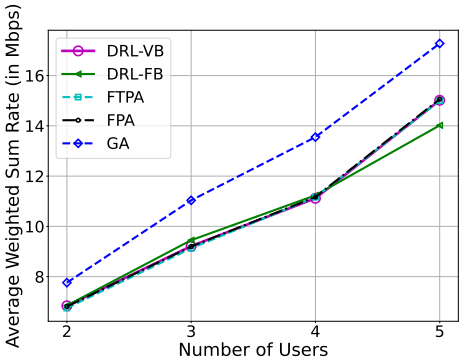

Fig. 1: Average achieved weighted sum rate in terms of K.

In Fig. 1, the average achieved WSR is plotted in terms of the number of users. Fig. 1 shows that, as expected, 'GA' outperforms all other methods at the cost of an additional computational complexity, as shown in Table II. All remaining methods perform similarly in terms of the achieved WSR.

In Fig. 2, the average percentage of satisfaction for both the user rate requirements and the SIC stability conditions with respect to the number of users K is plotted. In terms of rate satisfaction, only the proposed DRL-method with variable bounds and the GA are able to achieve 100% satisfaction for all K values. In terms of satisfaction of the SIC stabil-

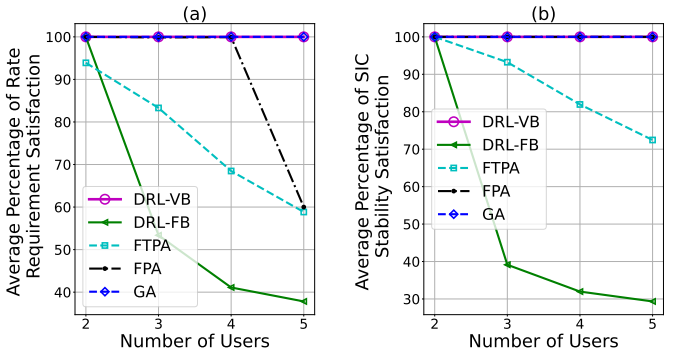

Fig. 2: (a) Average percentage of rate requirements satisfaction in terms of K , (b) Average percentage of SIC stability conditions satisfaction in terms of K.

ity conditions, 'DRL-VB', 'GA' and 'FPA' achieve a 100% satisfaction while the performance of 'DRL-FB' and 'FTPA' rapidly degrades when K increases. It should be noted that for 'DRL-FB', when $K = 5$, the number of available power levels L is set to 5 as otherwise, the SIC stability satisfaction percentage would be 0. The number of available power levels L is set to 4 for all other cases. In fact, system complexity increases with the number of available power actions at the BS level, given by L^K. Hence, a smaller value of L is preferred as it results in a lower complexity.

Finally in Table II, the mean time needed (in seconds), using an Intel I7 $-$ 4790 processor clocked at 3.6 GHz, to find the power allocation solution is shown for the different methods. FTPA and FPA require the least amount of time thanks to the use of simple algebraic solutions to solve the power allocation problem. However, the solutions yielded by both FTPA and FPA do not satisfy the rate requirements and SIC stability constraints for all K values as previously shown in Fig. 2. Table II shows that the GA is, by far and with several orders of magnitude difference, the slowest to find a solution that satisfies the different requirements. On the other hand, the DRL-VB proposed method results in a very manageable time complexity, requiring only an average of 0.2% the time needed by the GA to find efficient power allocation solutions that satisfy all system constraints.

TABLE II: Average time needed by the different methods (s)

	$K=2$	$K=3$	$K=4$	$K=5$
DRL-VB	1.17×10^{-2}	1.19×10^{-2}	3.02×10^{-2}	6.01×10^{-2}
DRL- FB	10^{-2}	1.01×10^{-2}	2.65×10^{-2}	5.7×10^{-2}
FTPA	6.67×10^{-5}	8.25×10^{-5}	10^{-4}	2.8×10^{-4}
FPA	4.64×10^{-5}	6.23×10^{-5}	8×10^{-5}	2×10^{-4}
GA	5.8	9.81	14.2	19.4

To conclude, the proposed DRL-based power allocation method is able to find solutions that satisfy rate and SIC stability constraints with a manageable time complexity. It can do so while achieving about 85% of the performance of the genetic method in terms of WSR, albeit with a much lower computational complexity. Moreover, the results prove the effectiveness of varying the search space bounds in the DRL-based method.

V. Conclusion

In this paper, a novel power allocation method based on deep reinforcement learning was introduced. With the aim of maximizing the weighted sum rate, the proposed method starts from a discrete set of possible power levels and progressively adapts the power values search space. Through this progressive refining of the search space, the proposed method is able to optimize the power allocation in a *quasi-continuous* manner, while keeping constant system complexity. The numerical results reveal that the proposed DRL-power allocation method outperforms two baseline power allocation algorithms in terms of satisfying target user rates and SIC stability conditions. Moreover, while providing similar satisfaction rates, it achieves more than 85% of the weighted sum rate obtained by a largely more complex genetic algorithm, making it the method of choice to solve such a problem.

References

[1] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, Y. Selén, and J. Sköld, "5G wireless access: requirements and realization," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 42–47, December 2014.

[2] M. J. Youssef, J. Farah, C. A. Nour, and C. Douillard, "Resource allocation in NOMA systems for centralized and distributed antennas with mixed traffic using matching theory," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 414–428, Jan. 2020.

[3] J. Zhu, J. Wang, Y. Huang, S. He, X. You, and L. Yang, "On optimal power allocation for downlink non-orthogonal multiple access systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2744–2757, Dec. 2017.

[4] J. Wang, Q. Peng, Y. Huang, H.-M. Wang, and X. You, "Convexity of weighted sum rate maximization in NOMA systems," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1323–1327, Sept. 2017.

[5] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.

[6] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, p. 279–292, May 1992.

[7] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, "Deep reinforcement learning for throughput improvement of the uplink grant-free NOMA system," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6369–6379, July 2020.

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

[9] H. Yang, Z. Xiong, J. Zhao, T. D. Niyato, C. Yuen, and R. Deng, "Deep reinforcement learning based massive access management for ultra-reliable low-latency communications," *IEEE Trans. Wireless Commun., Early Access*, pp. 1–1, 2020.

[10] S. Wang, T. Lv, W. Ni, N. C. Beaulieu, and Y. Jay Guo, "Joint resource management for MC-NOMA: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun., Early Access*, pp. 1–1, Apr. 2021.

[11] J. G. Andrews, "Interference cancellation for cellular systems: a contemporary overview," *IEEE Wireless Commun.*, vol. 12, no. 2, pp. 19–29, April 2005.

[12] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, "On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users," *IEEE Signal Process. Lett.*, vol. 21, no. 12, pp. 1501–1505, Dec. 2014.

[13] "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529–533, Feb. 2015.

[14] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, 2019.

[15] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3668–3681, Aug. 2020.

[16] S. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Springer-Verlag Berlin Heidelberg, 2008.

[17] K. I. Ahmed and E. Hossain, "A deep Q-learning method for downlink power allocation in multi-cell networks," 2019. [Online]. Available: http://arxiv.org/abs/1904.13032