Cooperative Reinforcement Learning for Adaptive Power Allocation in Device-to-Device Communication

Muhidul Islam Khan*, Muhammad Mahtab Alam*, Yannick Le Moullec* and Elias Yaacoub†
*Thomas Johann Seebeck Department of Electronics
Tallinn University of Technology
Ehitajate tee 5, 19086 Tallinn, Estonia
Email: mdkhan@ttu.ee, muhammad.alam@ttu.ee, yannick.lemoullec@ttu.ee
†Faculty of Computer Studies
Arab Open University, Beirut, Lebanon.
Email: eliasy@ieee.org

***Abstract*—Mobile devices are an intrinsic part of the Internet of Things (IoT) paradigm. Device-to-device (D2D) communication is emerging as one of the viable solutions for the radio resource optimization in an IoT infrastructure. However, it also comes with the challenges associated with power allocation as it causes severe interference by reusing the spectrum with the cellular users in an underlay model. Therefore, efficient techniques are required to reduce the interference with proper power allocation. In this paper, we propose a cooperative reinforcement learning algorithm for adaptive power allocation in D2D communication which helps to provide better system throughput as well as D2D throughput with less interference. We perform cooperation by sharing the value function between devices and incorporating a neighboring factor. We design our states for reinforcement learning with appropriate application-defined variables which provide a longer observation space. We compare our work with the existing distributed reinforcement learning method and random allocation of resources. Simulation results show that the proposed algorithm outperforms the distributed reinforcement learning and the random allocation both in terms of overall system throughput as well as D2D throughput by adaptive power allocation.**

I. Introduction

The advancement of short-range communication networks and the seamless interconnection between devices pave the way for massive machine type communication in Internet of Things (IoT). Devices will be the main components in the Internet of Things (IoT) paradigm comparable to the way humans use the internet. Massive machine type communication is considered as an important feature offered by fifth generation (5G) cellular networks to enhance IoT services [1]. However, this massive machine type communication poses some requirements on the network, e.g., increased system throughput by improving spectrum efficiency, long device battery life and quality of service (QoS). Device-to-Device (D2D) communication, one of the key technical features of 5G communications, provides an efficient way to cope with these requirements [2].

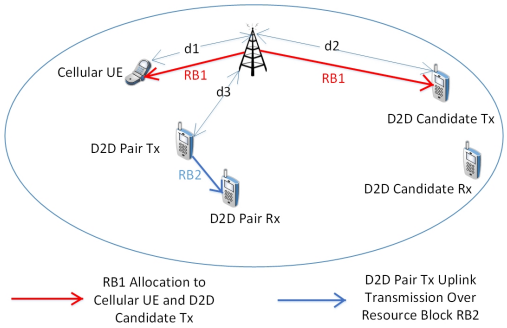


Fig. 1. D2D Communication in Cellular Network

D2D communication is proposed in Long Term Evolution Advanced (LTE-Advanced) standard to facilitate increased system throughput and to reduce traffic load in the core network. D2D communication operates in two modes: overlay and underlay where the D2D devices share the resources with the cellular users (CU) in a non-orthogonal or an orthogonal way respectively [3]. One of the key features of D2D communication is that it can enhance the network capacity to a great extent by reusing the spectrum between D2D and cellular user. However, D2D devices generate interferences while reusing the spectrum. This interference can be controlled if proper power levels are allocated to the D2D devices. Although higher transmission power can increase D2D throughput, it increases interference level as well. Therefore, choosing the proper level of transmission power is a crucial research issue in D2D communication.

To illustrate the problem, Figure 1 shows a basic single cell scenario with one cellular user (CU), two D2D pairs and one base station (BS) having two resource blocks

(RBs) operating in an underlay mode. D2D devices contend for resource blocks (RB) for reusing. Here, RB1 is allocated to the cellular user. D2D pair Tx and D2D pair Rx are assigned RB2. Now, D2D candidate Tx and D2D candidate Rx will contend for the resources either for RB1 or for RB2 to access. If we allocate RB1 to a D2D pair closer to the BS, there will be high interference between the D2D pair and the cellular user. So, RB1 should be allocated to the D2D candidate Tx which is close to the cell edge. For reusing the RB1, there will be interferences. It is possible to minimize the level of interferences and maximize the throughput of the system by selecting proper level of transmission power for D2D communication by a learning algorithm and which is the main goal of our paper.

In this work, we apply a cooperative reinforcement learning algorithm for the power allocation by scheduling the actions in D2D communication. The power allocation will be in a way that the overall system throughput is increased and at the same time the QoS of cellular users is maintained by keeping a proper level of interference. For reinforcement learning, we consider an on-policy learning algorithm which helps to learn scheduling of actions based on exploration and exploitation. We consider a set of actions based on the level of power allocation for the D2D users. Cooperation between devices is considered in this learning by means of sharing the value function with each other and considering a neighboring factor which helps to increase the observation space for power allocation. Moreover, our reward function comprises of interference level, QoS of the cellular users, and the channel gains between the devices.

II. Related Works

Recent advances in reinforcement learning (RL) create a broad scope of adaptive applications to apply. Power allocation in D2D communication is such an application.

Esmat et al. [4] propose an adaptive power allocation method based on a two phase optimization algorithm. The initial set of candidate channels which can be reused by D2D devices is determined in the first phase. In the second phase, an optimization algorithm is used to determine the optimal power for D2D devices that increase the system throughput. Their proposed Lagrangian dual decomposition is computationally complex. In [5], the authors propose a heuristic power allocation algorithm. They design an interference restricted region where a cellular resource is shared by multiple D2D pairs. Both D2D and cellular devices make probabilistic resource sharing decisions which helps to increase the system throughput. Their method is not adaptive in terms of power allocation to the users.

Preliminary exploitation of machine learning techniques in D2D resource optimization were conducted by Luo et al. [6] and Nie et al. [7]. In [6], Luo et al. applied distributed Q-learning for dynamic resource allocation which improves overall system capacity in comparison to random allocator and baseline reference with maximum transmit power. However, in the modeling of reinforcement learning, the set of states and set of actions are not clearly specified. The action selection strategy at each time step is not considered. The reward function is designed by considering the signal to interference plus noise power ratio (SINR), but the channel gain between the base station and the user, the channel gain between users are not included. Channel gains are important to consider as these help to the D2D communication with better SINR level and transmission power which reflects in increased system throughput [8]. Nie et al. recently, [7] applied distributed Q-learning for the power control of the D2D users. They update the Q-values in both distributed and collaborative fashion. Their proposed approach outperforms the random allocation of resources, but the channel gains are not associated in their reward function.

Our proposed method helps to learn adaptively the appropriate level of transmission power for D2D communication which provides increases level of system throughput assuring the QoS of the network.

III. System Model

We consider a system model that consists of a set of cellular users i.e., $CU = \{1, 2, \ldots, C\}$ and a set of D2D pairs i.e., $D = \{1, 2, \ldots, D\}$. We assume that there are R number of resource blocks (RB). We consider the uplink transmission where both cellular users and D2D users share the same available RBs. Each cellular user occupies one RB which can be shared by D2D pairs. Now, this may cause several interferences: (i) BS receives from D2D transmitter (D2D Tx), (ii) D2D receivers (D2D Rx) receive from cellular users and (iii) D2D receivers (D2D Rx) receive from D2D transmitters in other D2D pairs. SINR can be treated as an important factor to measure the link quality. We can denote the D2D user's SINR on the rth RB as follows:

$$\gamma_r^{D_u} = \frac{p_r^{D_u}.G_{D_{u,r}}^{uu}}{\sigma^2 + p_r^c.G_r^{c_u} + \sum_{v \in D_r}^{v \neq u} p_r^{d_v}.G_{D_{v,r}}^{uv}} \quad (1)$$

where $p_r^{D_u}$ and p_r^c denote the uth D2D user and cellular user uplink transmission power on rth RB, respectively. $p_r^{D_u} \leq P_{max}$, $\forall_u \in D$ where P_{max} is the upper bound of each D2D user's transmit power. σ^2 is the noise variance [9].

$G_{D_{u,r}}^{uu}$, $G_{D_{v,r}}^{uv}$ and $G_r^{c_u}$ are the channel gains in the uth D2D link, channel gain from D2D transmitter u to receiver v, and the channel gain from cellular transmitter c to receiver u, respectively. D_r is a D2D pairs set sharing the rth RB.

The SINR of a cellular user $c \in CU$ on the rth RB is

$$\gamma_r^c = \frac{p_r^c.G_{c,r}}{\sigma^2 + \sum_{v \in D_r} p_r^{d_v}.G_{v,r}} \quad (2)$$

where $G_{c,r}$ and $G_{v,r}$ indicate the channel gains on the rth RB from BS to cellular user c and vth D2D transmitter, respectively.

The total path-loss which includes the antenna gain between BS and the user u is:

$$PL_{dB,B,u(.)} = L_{dB}(d) + \log_{10}(X_u) - A_{dB}(\theta) \quad (3)$$

where $L_{dB}(d)$ is the pathloss between a BS and the user at a distance d meter. X_u is the lognormal shadow pathloss of user u. $A_{dB}(\theta)$ is the radiation pattern.

$L_{dB}(d)$ can be expressed as follows:

$$\begin{aligned} L_{dB}(d) = 40(1 - 4 \times 10^{-3} h_b) \log_{10}(d/1000) \\ -18 \log_{10}(h_b) + 21 \log_{10}(f_c) + 80 \end{aligned} \quad (4)$$

where f_c is the carrier frequency in GHz and h_b is the base station antenna height [10]. The linear gain between the BS and a user is $G_{Bu} = 10^{\frac{-PL_{dB,B,u}}{10}}$ [11].

For D2D communication, the gain between two users u and v is $G_{uv} = K_{uv} d_{uv}^{-\alpha}$. Here, d_{uv} is the distance between transmitter u and receiver v. α is a constant pathloss exponent and k_{uv} is a normalization constant.

We mainly focus on the system performance, particularly to enhance the simultaneous throughput of both D2D users and cellular users. The power allocation of D2D users can be obtained by solving the following equation:

$$\begin{aligned} &\max_{p_r} \sum_{r=1}^{R} \{\log_2(1 + \gamma_r^c) + \sum_{u \in D_r} \log_2(1 + \gamma_r^{D_u})\} \\ &\text{subject to } \gamma_r^c \geq \tau_0 \\ &0 \leq p_r^{D_u} \leq P_{max}, \forall_{u,r} \end{aligned} \quad (5)$$

where $p_r^{D_u} = (p_r^1, p_r^2, \ldots, p_r^D)$. The objective function is to maximize the global throughput. In general, D2D throughput increases with the increase of transmission power, at an expense of higher interference at cellular users. Our goal is apply cooperative reinforcement learning and to investigate the optimal transmit power p_r^D in such a way that the D2D throughput is maximized without compromising the QoS of the cellular users.

IV. Cooperative Reinforcement Learning Algorithm for Resource Allocation

We apply reinforcement learning algorithm named state action reward state action, SARSA(λ), for power allocation in D2D communication. This variant of standard SARSA(λ) algorithm [12] has some important features like cooperation by using neighboring weight factor, a heuristic policy for exploration and exploitation, and a varying learning rate considering the visited state-action pair. We consider the cooperative fashion of this learning algorithm which helps to improve the throughput as explained in Section I by sharing the value function and incorporating weight factors for the neighbors of each agent. In reinforcement learning, there is no need of prior knowledge about the environment. Agents learn how to behave with the environment based on the previous experience achieved so far, which is traced by Q-value and controlled by a reward function. There are some actions/tasks to perform at every time step. After performing every action, the agents shifts from one state to another and it gets a reward that reflects the impact of that action, which helps to decide about the next action to perform.

In our environment, we consider the components of the reinforcement learning algorithm as follows:

Agent: All the resource allocators: Base stations, D2D Transmitters.

State: The state of D2D user u on RB r at time t is defined as:

$$S_t^{u,r} = \gamma_r^c \cup G_{Bu} \cup G_{uv}$$

We consider three variables γ_r^c, G_{Bu} and G_{uv} for defining the states. γ_r^c is the SINR of a cellular user on the rth RB. G_{Bu} is the channel gain between the BS and an user u. G_{uv} is the channel gain between two users u and v.

Now these variables can be either 0 or 1. $\gamma_r^c \geq \tau_0$, $G_{Bu} \geq \tau_1$ and $G_{uv} \geq \tau_2$ means the state value 1 and $\gamma_r^c < \tau_0$, $G_{Bu} < \tau_1$ and $G_{uv} < \tau_2$ means the state value 0. In this way, the total number of possible states is 8 where τ_0, τ_1 and τ_2 are the minimum SINR and channel gain guaranteeing the QoS performance of the system.

Action/Task: The action of each agent consists of a set of transmitting power levels. It is denoted by

$$A = (a_1^r, a_2^r, \ldots, a_{pl}^r)$$

where r represents the rth Resource Block (RB), and pl means that every agent has pl power levels. In this work, we consider some fixed power levels to assign within the range of 1 to P_{max} in the interval 3 dBm.

Reward Function: The system average capacity on the RB is according to the following equation after the agent performs action:

$$\Re = \frac{\sum_{u=1}^{U} \log_2(1 + SINR(u))}{U} \quad (6)$$

when $\gamma_r^c \geq \tau_0$, $G_{Bu} \geq \tau_1$ and $G_{uv} \geq \tau_2$. Otherwise, $\Re$= - 1. U denotes the number of the users in the cell, SINR (u) denotes the signal to interference plus noise power ratio of user u. Here, we assume that the channel gain information are obtained by the centralized controller, i.e., the base station.

SARSA(λ) is an on-policy reinforcement learning algorithm that estimates the value of the policy being followed where λ is a parameter such as learning

rate [13]. In SARSA(λ) learning algorithm, every agent needs to maintain a Q-matrix like Q-learning [14] which is initially assigned 0 and the agents may be in any state. Based on performing one particular action, it shifts from one state to another. The basic form of the learning algorithm is $(s_t, a_t, \Re, s_{t+1}, a_{t+1})$, which means that the agent was in state s_t, did action a_t, received reward $\Re$, and ended up in state s_{t+1}, from which it decided to perform action a_{t+1}. This provides a new iteration to update $Q(s_t, a_t)$.

SARSA(λ) helps to find out the appropriate action for a particular state. The considered state-action pair's value function $Q_t(s_t, a_t)$ as follows:

$$Q_t(s_t, a_t) = \Re + \Gamma Q_{t+1}(s_t, a_t) \tag{7}$$

In Equation 7, Γ is a *discount-factor* which varies from 0 to 1. The higher the value, the more the agent relies on future rewards than on the immediate reward. The objective of applying reinforcement learning is to find the optimal policy $Q_t^{\pi}(s_t, a_t)$ which maximizes the value function $\pi = \max_{\pi} Q_t^{\pi}(s_t, a_t)$. We consider the cooperative fashion of this algorithm where each agent shares the value function with each other.

At each time step, Q_{t+1} for the iteration $t+1$, Q_{t+1} is updated with the temporal difference error δ_t and the immediate received reward. The Q value has the following update rules:

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha \delta_t e_t(s_t, a_t) \tag{8}$$

for all s, a.

In Equation 8, $\alpha \in [0, 1]$ is the learning rate which decreases with time. δ_t is the temporal difference error which is calculated by following rule:

$$\delta_t = \Re_{t+1} + \Gamma f Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \tag{9}$$

$\Re_{t+1}$ represents the reward received for performing an action. f is the *neighboring weight factor* of agent i and is defined as follows:

$$f = \frac{1}{ngh(n_i)} \qquad \text{if } ngh(n_i) \neq 0 \tag{10}$$

$$= 1 \qquad \text{otherwise.} \tag{11}$$

where $ngh(n_i)$ is the number of neighbors of an agent, i.

There is a trade-off between exploration and exploitation in reinforcement learning. Exploration chooses an action randomly in the system to find out the utility of that chosen action. Exploitation deals with the actions which have been chosen based on previously learned utility of the actions.

We use a heuristic for exploration probability at any given time such as:

$$\epsilon = \min(\epsilon_{max}, \epsilon_{min} + k * (S_{max} - S)/S_{max}) \tag{12}$$

where ϵ_{max} and ϵ_{min} denote upper and lower boundaries for the exploration factor, respectively. S_{max} represents the maximum number of states which is eight in our work and S represents the current number of states already known. At each time step, the system calculates ϵ and generates a random number in the interval $[0, 1]$. If the selected random number is less than or equal to ϵ, the system chooses a uniformly random task (exploration), otherwise it chooses the best task which gives maximum Q value (exploitation). k is a constant which controls the effect of unexplored states.

Eligibility trace helps to improve the learning technique in SARSA(λ). In Equation 13, $e_t(s_t, a_t)$ is the eligibility trace. Here, λ is learning parameter for guaranteed convergence, whereas, Γ_1 is the discount factor. In addition, the eligibility trace helps to provide higher impact on revisited states. For example, for a state-action pair (s_t, a_t), if $s_t \in s$ and $a_t \in a$, the state-action pair is reinforced. Otherwise, the eligibility trace is removed.

The eligibility trace is updated by the following rule:

$$e_t(s_t, a_t) = \Gamma_1 \lambda e_{t-1}(s_t, a_t) + 1 \quad \text{if } s_t \in s \text{ and } a_t \in a$$
$$e_t(s_t, a_t) = \Gamma_1 \lambda e_{t-1}(s_t, a_t) \qquad \text{otherwise.} \tag{13}$$

Algorithm 1 Proposed power allocation method.

1: Initialize all the parameters based on Table I
2: **loop**
3: Calculate the channel gains by considering the pathloss (Eq. 4)
4: Calculate the SINR of the D2D users on the $r$the RB (Eq. 1)
5: Calculate the SINR of the cellular users on the RB (Eq. 2)
6: **if** ($\gamma_r^c \geq \tau_0$, $G_{Bu} \geq \tau_1$ and $G_{uv} \geq \tau_2$) **then**
7: $\Re = \frac{\sum_{u=1}^{U} \log_2(1+SINR(u))}{U}$
8: **else**
9: $\Re$=-1
10: Apply Algorithm 2 for the power allocation
11: **end if**
12: **end loop**

The learning rate α is decreased in such a way that it reflects the degree to which a state-action pair has been chosen in the recent past. It is calculated as:

$$\alpha = \frac{\rho}{visited(s, a)} \tag{14}$$

where ρ is a positive constant and $visited(s, a)$ represents the visited state-action pairs so far.

Algorithm 1 shows how the proposed power allocation method works. Algorithm 2 shows how the variant of SARSA(λ) works.

Algorithm 2 Execution steps of cooperative SARSA(λ) reinforcement learning algorithm over number of iterations.

1: Initialize $Q(s,a)$=0, $e(s,a)$=0, ϵ_{max}=0.3, ϵ_{min}=0.1, k=0.25, ρ=1, $\Gamma = 0.9$, $\Gamma_1 = 0.5$, $\lambda = 0.5$
2: **loop**
3: Determine the current state s based on γ_r^c, G_{Bu} and G_{uv}
4: Select a particular action a based on the policy (Eq. 12)
5: Execute the selected action
6: Update learning rate (Eq. 14)
7: Determine the temporal difference error (Eq. 9)
8: Update eligibility traces
9: Update the Q-value (Eq. 8)
10: Update the value function and share with neighbors
11: Shift to the next state based on the executed action
12: **end loop**

V. Simulation Results

We implement our proposed cooperative reinforcement learning algorithm and compare it with the random allocation and existing distributed reinforcement learning algorithm.

TABLE I
Simulation Parameters.

Parameter	***Value***
P_{max}	23 dBm
Number of resource blocks	30
Number of cellular users	30
Number of D2D user pairs	12
D2D radius	20 m
Pathloss parameter	3.5
Cell radius	500 m

We consider a single cell with a radius of 500 m where some cellular users and D2D pairs are uniformly distributed within the coverage of BS. The parameters for the simulation are shown in Table I.

We consider τ_0=0.004, τ_1=0.2512 and τ_2=0.2512 as constraints to define the states [15]. In our reinforcement learning algorithm, we consider ϵ_{max}=0.3, ϵ_{min}=0.1 and k=0.25. The constant for learning rate update, ρ=1 and the discount factor, $\Gamma = 0.9$ are considered based on the work [7] for fair comparison with our work.

We compare our method with the distributed reinforcement learning proposed in [7] and random allocation of resources. Figure 2 shows that the proposed cooperative reinforcement learning outperforms both the random allocation and the distributed reinforcement learning regarding average system throughput calculated by the Equation 5 considering 12 D2D user pairs and other parameter values as in Table I. We can observe that after

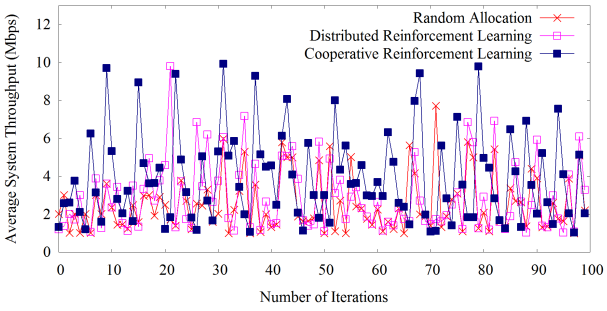
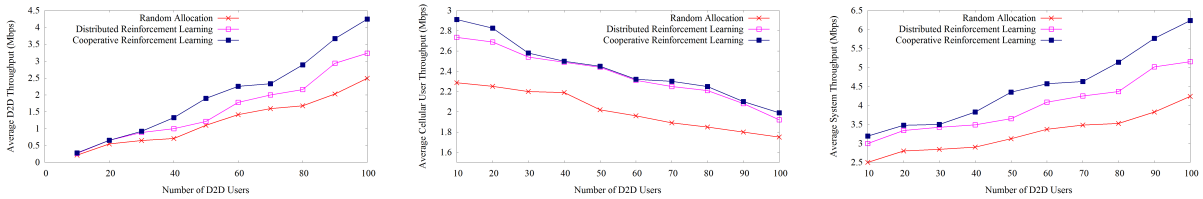


Fig. 2. Average system throughput over the number of iterations

the 30th iteration, our proposed learning algorithm outperforms other methods at almost every iteration when the algorithm reaches the convergence of learning. There are some points where distributed reinforcement learning outperforms proposed cooperative reinforcement learning due to the fact that we consider the heuristic action selection policy based on exploration and exploitation in Equation 12 which avoids to stuck the learning algorithm in a local optimum. Random allocation shows poor results since it does not act appropriately with the changes of the environment. Whereas, distributed reinforcement learning show moderate results comparing with the both methods. We consider 100 iterations here for the comparison, but the trend of outperformance of our proposed algorithm remains the same with additional iterations.

Figure 3 shows the average D2D throughput, average cellular user throughput and the average system throughput over the number of D2D users. We can observe that D2D throughput increases with the increase of D2D users, but on the other hand, cellular user throughput decreases. System throughput is the summation of D2D and the cellular user throughput, which also increases with the increment of the D2D users. All methods show these same trends over the number of D2D users. From this experiment, we can also investigate the issue about the appropriate number of D2D users which provides the better trade-off between D2D and cellular user throughput in a single cell scenario. Here, we can observe that moderate number of D2D users, for example, 50 D2D users provide suitable amount of D2D and cellular user throughput. Our proposed method outperforms the other methods regarding D2D throughput, cellular user throughput and overall system throughput at every number of D2D users.

Figure 4 shows the average D2D throughput over transmit power applying our proposed method, distributed reinforcement learning and random allocation of resources. All the methods follow the same trend that with the increase of transmit power, D2D throughput increases. Our proposed reinforcement learning outperforms others at every level of transmit power due to the



(a) Average D2D user throughput over number of D2D users (b) Average cellular user throughput over number of D2D users (c) Average system throughput over number of D2D users

Fig. 3. Throughput analysis over number of D2D users

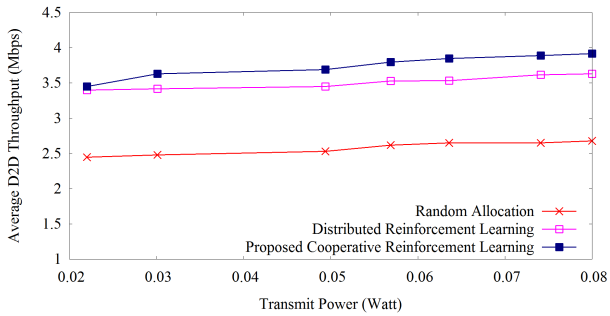


Fig. 4. D2D throughput versus transmit power

appropriate learning of transmission power assignment to the resource blocks. Our proposed method increases D2D throughput by 6.2% as compared with the distributed reinforcement learning. On the other hand, random allocation shows the lowest performance as usual due to the allocation of resources without adaptiveness.

VI. Conclusion

Adaptive power allocation is critical for improving the throughput in the context of D2D communication. Reinforcement learning can be considered as a suitable method for the adaptive power allocation by scheduling the actions performed by the resource allocators. In this work, we apply a cooperative reinforcement learning for the action scheduling where the actions are to assign different levels of transmission power. Our method is compared with the distributed reinforcement learning and random allocation of resource. The results show better performance in terms of system throughput as well as D2D throughput.

Currently, we consider a single cell setup, for the future work, we will consider multiple cells considering each cell as an agent in a distributed fashion which will fully use the benefits of reinforcement learning.

Acknowledgment

This research was supported by the Estonian Research Council through the Institutional Research Project IUT19-11, and by the Horizon 2020 ERA-chair Grant "Cognitive Electronics COEL"-H2020-WIDESPREAD-2014-2 (Agreement number: 668995; project TTU code VFP15051).

References

[1] L. Ji, B. Han, M. Liu, and H. D. Schotten, "Applying device-to-device communication to enhance iot services," *arXiv preprint arXiv:1705.03734*, 2017.

[2] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, 2016.

[3] Z. Liu, T. Peng, S. Xiang, and W. Wang, "Mode selection for device-to-device (d2d) communication under lte-advanced networks," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 5563–5567.

[4] H. Esmat, M. M. Elmesalawy, and I. Ibrahim, "Adaptive resource sharing algorithm for device-to-device communications underlaying cellular networks," *IEEE Communications Letters*, vol. 20, no. 3, pp. 530–533, 2016.

[5] F. Hajiaghajani and M. Rasti, "An adaptive resource allocation scheme for device-to-device communication underlaying cellular networks," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2015, pp. 1–6.

[6] Y. Luo, Z. Shi, X. Zhou, Q. Liu, and Q. Yi, "Dynamic resource allocations based on q-learning for d2d communication in cellular networks," in *IEEE International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2014, pp. 385–388.

[7] S. Nie, Z. Fan, M. Zhao, X. Gu, and L. Zhang, "Q-learning based power control algorithm for d2d communication," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2016, pp. 1–6.

[8] Y. Hwang, J. Park, K. W. Sung, and S.-L. Kim, "On the throughput gain of device-to-device communications," *ICT Express*, vol. 1, no. 2, pp. 67–70, 2015.

[9] M. Zulhasnine, C. Huang, and A. Srinivasan, "Efficient resource allocation for device-to-device communication underlaying lte network," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2010, pp. 368–375.

[10] F. Graziosi and F. Santucci, "A general correlation model for shadow fading in mobile radio systems," *IEEE Communications Letters*, vol. 6, no. 3, pp. 102–104, 2002.

[11] M. Zulhasnine, C. Huang, and A. Srinivasan, "Penalty function method for peer selection over wireless mesh network," in *IEEE Vehicular Technology Conference Fall (VTC 2010-Fall)*, 2010, pp. 1–5.

[12] M. I. Khan, "Resource-aware task scheduling by an adversarial bandit solver method in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 10, 2016.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[14] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[15] R. Wang, J. Zhang, S. Song, and K. B. Letaief, "Optimal qos-aware channel assignment in d2d communications with partial csi," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7594–7609, 2016.