Distributed Deep Reinforcement Learning Resource Allocation Scheme For Industry 4.0 Device-To-Device Scenarios

Romero, Jesus Burgueno; Adeogun, Ramoni Ojekunle; Bruun, Rasmus; Morejon, Santiago; de-la-Bandera, Isabel ; Barco, Raquel

Published in:
IEEE Vehicular Technology Conference - Fall (VTC-Fall)

Publication date:
2021

Distributed Deep Reinforcement Learning Resource Allocation Scheme For Industry 4.0 Device-To-Device Scenarios

Jesús Burgueño[2], Ramoni Adeogun[1], Rasmus Liborius Bruun[1] C. Santiago Morejón García[1]
Isabel de-la-Bandera[2] Raquel Barco[2]
[1] *Department of Electronic Systems, Aalborg University, Denmark*
[2] *Instituto Universitario de Investigación en Telecomunicación (TELMA), Universidad de Málaga, CEI Andalucía TECH, Málaga, Spain*
E-mail: jesusbr@ic.uma.es, {ra, rlb, csmg}@es.aau.dk, ibanderac@ic.uma.es, rbm@ic.uma.es

***Abstract*—This paper proposes a distributed deep reinforcement learning (DRL) methodology for autonomous mobile robots (AMRs) to manage radio resources in an indoor factory with no network infrastructure. Hence, deep neural networks (DNN) are used to optimize the decision policy of the robots, which will make decisions in a distributed manner without signalling exchange. To speed up the learning phase, a centralized training is adopted in which a single DNN is trained using the experience from all robots. Once completed, the pre-trained DNN is deployed at all robots for distributed selection of resources. The performance of this approach is evaluated and compared to 5G NR sidelink mode 2 via simulations. The results show that the proposed method achieves up to 5% higher probability of successful reception when the density of robots in the scenario is high.**

***Index Terms*—Industry 4.0, deep reinforcement learning, device-to-device, resource allocation, decentralized communications**

I. Introduction

Factories of Future with autonomous robots in charge of the manufacturing is one of the most important verticals targeted by the fifth generation (5G) of mobile communication systems in the Industry 4.0 domain. This environment enables huge benefits for manufacturing companies, such as the flexibility and adaptability of a factory to new and completely different production procedures, as well as the ability to make dynamic changes to ongoing production. In order to achieve these advantages and to perform proper production, the robots must be well coordinated with each other. This implies that one of the most relevant issues is the communication between robots, as often a large number of robots move around the factory and need to exchange information in real time. Hence, managing radio time-frequency resources for communication is one of the most challenging points [1], [2]. Some factories tackle this issue with an external network infrastructure that takes decisions centrally [3]. In that case, each network node could make decisions for each of the robots it serves. However, a dedicated communication infrastructure is not always guaranteed, since it is necessary to install base stations. In addition, the network equipment may stop working properly necessitating a scenario where each robot makes its own decisions autonomously in order for manufacturing to continue. Thus, device-to-device (D2D) communication is required to allow robots communicate with each other without network assistance.

As an alternative solution, devices can manage the radio resources in a decentralized manner. One or more robots can be in charge of allocating the resources to be used by each of the neighboring robots. However, this would be problematic in dense scenarios where a large number of robots share a common resource pool due to the vast amount of signalling traffic that would have to be exchanged. On the other hand, if resource allocation is completely distributed and no signalling is exchanged between devices, the lack of coordination can lead to data collisions in case two or more devices access the same resource simultaneously. To tackle this issue, deep reinforcement learning (DRL) techniques allow devices to experience multiple solutions for several different situations in the scenario, enabling the devices to explore which will produce the best results. Therefore, robots could use a distributed DRL resource allocation scheme to avoid coordination by signalling exchange, and acquire this coordination ability intrinsically in their decision policy by prior training.

To perform decision policy training in scenarios with a large number of devices, a popular approach proposes to train a single agent but using complete information of the scenario which is collected from all devices [4], [5]. However, it is difficult to obtain such information in real scenarios with communication constraints, since a device must gather information from all distributed devices in real time [6]. Similarly, a centralized node uses DRL to manage the subcarriers and channels in a D2D environment underlaying cellular networks [7], [8]. In contrast, this work proposes a distributed methodology for non network-assisted D2D scenarios. In this sense, only the scenario partial view of a single robot is used to train its decision policy, so only the information available at each robot is used without the need for ideal signalling exchange as in [4], [5]. However, it should be remarked that a single robot will perform a centralized training with the individual experiences of all robots to accelerate the training. Once completed, the pre-trained decision policy is deployed at each robot, which

will be able to make a decision based only on its partial view of the scenario. In summary, the key contributions of this paper are:

- Design of a distributed DRL solution to the problem of autonomous resource selection in non network-assisted D2D scenarios.
- The proposed approach allows training to be performed on each robot individually, as they only use their particular point of view to make decisions. However, a training strategy where a single deep neural network (DNN) is trained with experience from all simulated robots is adopted to speed up the training. The pre-trained DNN is then deployed to all robots to carry out the performance tests in a distributed manner.
- Comparison of the performance of the DRL solution and 3GPP release 16 NR sidelink mode 2 with different deployment densities.

The paper is organized as follows: Section II details the use case addressed in this study. Section III describes the DRL methodology proposed by the authors. Section IV discusses the simulation assumptions and the results analysis. The paper concludes with final remarks and future work in Section V.

II. Use Case

A. Use Case

In this use case initially introduced in [9], the robots are assumed to be performing their manufacturing tasks moving around a rectangular indoor factory with 120 x 50 m^2 of area [10]. No communication infrastructure is assumed, thus robots must manage the resource allocation autonomously. They communicate with each other based on proximity. For that purpose it is assumed two device-centric areas with different cooperation ranges: critical cooperation range (CCR) and extended cooperation range (ECR), as stated in [9]. Robots share video streams data messages each 10 ms within the CCR to provide collective perception of the environment to avoid collisions among them and with obstacles while performing the assigned tasks. Since the CCR is only 5 meters, robots will transmit with a power of 0 dBm. On the other hand, they exchange discovery messages with the position and heading direction within the ECR, which is 25 meters. It is assumed that this communication is ideally performed in a separate resource pool. In addition, it should be pointed out that the channel is modeled as established for an indoor factory with sparse clutter and low base station height (InF-SL) by 3GPP in [10].

B. NR sidelink mode 2

To enable such decentralized communications 3GPP proposes NR sidelink mode 2 [11]. This method indicates the time slot to be used from a complete frame, which consists of 40 time slots with a duration of 250 μs each because a numerology of 2 is assumed, as it provides the maximum amount of time slots within a frame for Frequency Range 1 [9].

NR sidelink mode 2 comprises three phases: sensing, selection and transmission. In the first phase, the devices sense the resource pool during a pre-configured observation period. In particular, they detect the reference signal received power (RSRP) from other devices that are performing semi-persistent (SPS) transmissions in the time slots [11]. This SPS resource selection proposes that devices to transmit in the same time slot for a pre-configured number of consecutive transmissions, so other devices can sense a less changing resource pool state. To define the number of consecutive SPS transmissions in the same time slot, an integer value in the interval $[5 \cdot \frac{100}{20}, 15 \cdot \frac{100}{20}]$ is randomly selected with equal probability in use cases with data transmissions periodicity lower than 20 ms [12]. Once devices sense the resource pool, NR sidelink mode 2 removes slots with RSRP above a dynamic threshold from the set of available slots in the selection phase. If the amount of available or suitable slots is less than 20%, the threshold is raised by 3 dB to increase the number of suitable slots, as indicated in [11]. Thus, this set of available slots will consist of the unoccupied or lowest RSRP slots of the entire frame. This allows to minimize the collision risk. Slots are randomly selected from the set in the last phase, such that the transmission can be performed with the required modulation and coding scheme (MCS). Once all scheduled SPS transmissions are transmitted, the device returns to the sensing phase in case other robots remain within the CCR to select a new slot, which can also be the same, depending on the current state of the resource pool. Moreover, it should be pointed out that a device cannot simultaneously listen to the transmission from other devices if it is transmitting, i.e., a half-duplex problem exists [3]. It should also be noted that robots need to be time-synchronized for NR sidelink mode 2 to work properly. To this end, a decentralized synchronization procedure from 5G NR is described in [13].

Figure 1 depicts the operation of NR sidelink mode 2. Robots 1 and 2 enter the CCR at time instant x, so each of them must select a time slot to perform SPS transmissions. Also note that the frame is relative to the time at which two robots enter the CCR, so they must select one of the next 40 time slots. Then, the robots keep moving and, suddenly, robots 2 and 3 discover each other within the CCR. Since robot 3 was not transmitting previously, it removes the slots in use by other devices and selects a free slot. If it selects an occupied slot, collisions could occur and it would not be able to listen to other devices while transmitting. Therefore, the performance of the NR sidelink mode 2 will be optimal as long as remaining slots are those used by distant robots. However, the slot selection will become more challenging as the scenario density increases. In this way, DRL is proposed to ameliorate problems associated with the NR sidelink mode 2 in dense scenarios via intelligent reuse of available time slots.

III. DRL Based Resource Allocation Method

This method proposes the deployment of a DNN within each robot in the scenario. This DNN selects the time slot to transmit, so it enables a distributed resource allocation scheme.

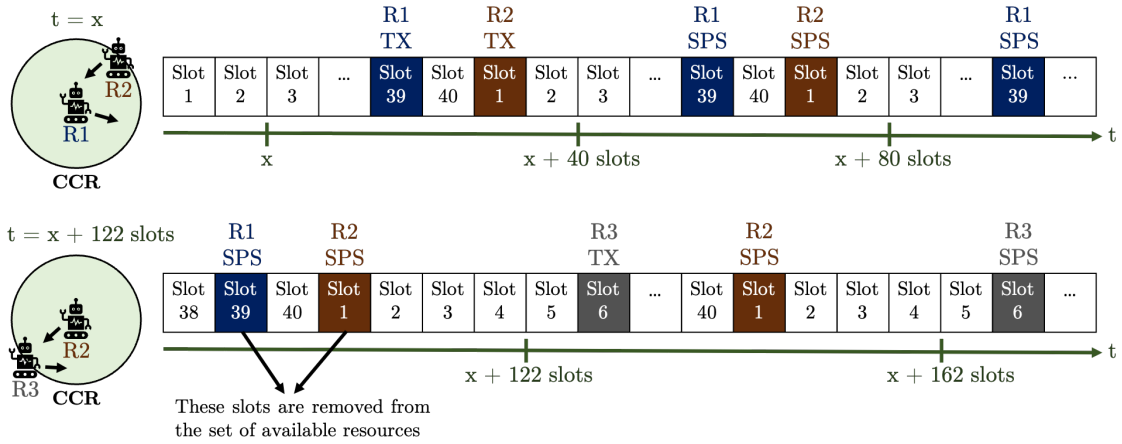

Fig. 1. Resource management by NR sidelink mode 2.

To speed up this decision making, the use of a DNN of small size is proposed: three hidden layers with 64 neurons each that use Rectified Linear Unit (ReLU) as activation function. A softmax output unit is then used at the last layer of the DNN. It calculates the probability of selecting each action using a Boltzmann distribution [14]. The higher the probability of being selected, the greater the expected reward of that action.

Before using this method, the DNN must be trained to allow robots to make the best decisions. With this purpose, three main elements are involved in the training of the decision policy. They are detailed below:

- **State (s).** Vector of RSRP values for each of the 40 time slots of the resource pool. These values are sensed during the pre-configured observation period before the robot discovers any other robot within the CCR. Therefore, the vector represents the robot's partial view of the environment, as the RSRP depends on its location in addition to transmissions from other robots. It should also be noted that no signalling exchange is required to define the state of the devices.
- **Action (a).** Selection of one of the 40 time slots for transmission. The DNN chooses this slot based on the sensed state. Once selected, a random number is generated in the range $[25, 75]$, as introduced in the previous section, to set the number of consecutive SPS transmissions in the same slot. This allows to decrease the variability of the resource pool state and the collision risk. If one or more neighboring devices remain within the CCR after completing all SPS transmissions, a new action is performed. This action can reuse the same slot or use a new one.
- **Reward (r).** Function that evaluates the state-action duple based on defined performance indicators and provides positive or negative feedback to train the decision policy. As multiple SPS transmissions are made for a single action, the reward relies on all of them. Likewise, transmissions will be multi-cast in case there are two or more robots within the CCR. Hence, the probability of successful reception is proposed to be maximized with the reward function (1). A negative reward is given for a SPS transmission in case it is received erroneously by any receiver. Hence, this function encourages a selection where every receiver is equally important, and avoid starvation of single receivers.

$$r = \frac{1}{N}\sum_{i=1}^{N} L_i \qquad L_i = \begin{cases} 1 & \text{if } C = R \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

L_i represents the contribution to r of each SPS transmission (i) and N denotes the number of SPS transmissions for a single action. On the other hand, R stands for the number of receivers within the CCR, whereas C indicates the number of them that correctly receive a single SPS transmission.

Figure 2 summarizes the interaction of the DNN of a single robot and the environment through these elements. It depicts several robots moving around a factory. When no robots are within the CCR and another one gets into it, each robot defines its state based on the previous observation period sensed. The devices then select a time slot to transmit and the reward is calculated once all SPS transmissions of a single decision have been completed. If many robots remain within the CCR, a new slot selection is made due to re-selection procedure as in NR sidelink mode 2. Finally, the DNN is updated when a set of experiences is collected, as detailed in next subsection.

A. Training Procedure

To achieve an optimal action selection, the weights (θ) of the DNN are updated throughout training using stochastic gradient ascent to maximize the reward obtained per decision [15]. θ is updated at the end of every episode once a batch of T (batch size) experiences is full. Each experience consists of a state (s_t), the action (a_t) performed for that state and the reward (r_{t+1}) obtained then. Thus, T defines the number of experiences used to calculate the gradient in each iteration. The smaller the parameter T the less accurate the gradient

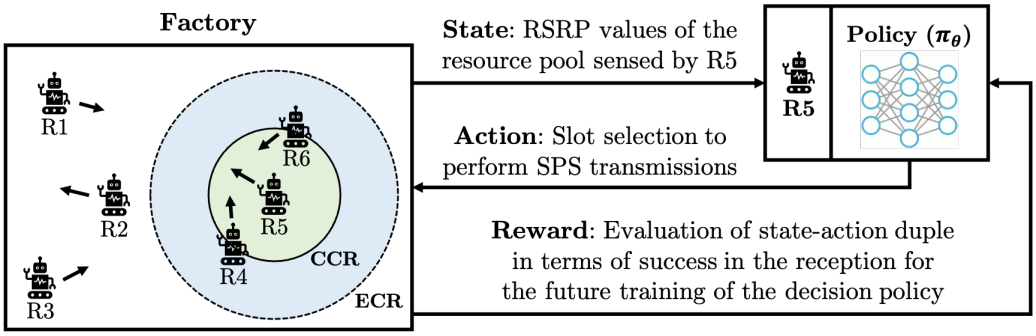

Fig. 2. Interaction between the decision policy of a single robot and the environment.

calculation will be. Conversely, the larger the parameter T the longer the training will be. Therefore, a trade-off between these two criteria will be adopted to set this value. On the other hand, each new episode generates a new scenario in which robots are randomly distributed throughout the factory, so a wider range of states will be experienced. The main goal of the training is to maximize the discounted cumulative future reward (G_t) obtained throughout each episode (2).

$$G_t = \sum_{t'=t+1}^{T} \gamma^{t'-t-1} r_{t'} \tag{2}$$

where γ is the discount factor, which is a value in the interval $[0, 1]$ that indicates the importance of future rewards to the current state. If the value is close to 0, only immediate rewards are considered. In contrast, long-term rewards will be considered if a value close to 1 is set. In this use case, a low value will be used to set γ since the long-term state of the scenario does not heavily depend on the current decision. Afterwards, the increment of θ with respect to the stochastic policy (π_θ), i.e. current θ values of the DNN, is computed. To simplify this step, the equation (3), which is derived analytically in [16], is used to update θ for the next episode. Since $\pi_\theta(a|s)$ indicates the probability of taking action a given state s with weights θ, this equation indicates that θ is updated in the direction in which the probability of obtaining the action that provides the highest reward increases the most:

$$\Delta\theta = \alpha \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) G_t \tag{3}$$

where α is the learning rate, which indicates the step size of each θ update. This parameter will be set to a very low value, as several states and actions must be explored before decision making converges.

The decision policy can take a long time to converge due to the large space of different states, as well as the great number of actions to choose for each of the states in this use case. This is why DRL approaches usually have a large state space but a small action space, which allows all possibilities to be explored in a shorter time. Nonetheless, the action space is also large in this use case. Thus, a reduction of the action space could drastically decrease the training time [17]. Therefore, we propose to reduce the action space to the first 20 slots relative to the moment when a robot discovers another one within the CCR. Still, all devices will be able to use all the slots in the frame over time, so devices' capabilities will not be restricted even if the action space is halved. In this sense, the entire frame could be used simultaneously by robots discovering others at different time points throughout the scenario. Furthermore, since only the first 20 time slots are considered for selection, the same slots will be used to create the state space. This will further reduce the convergence time.

In addition to reduce the state space, a further decrease in training time can be achieved by discretizing the continuous values [17]. In this way, the RSRP of each slot can be classified into different discrete levels. The smaller the number of levels, the fewer the number of different states to explore, which implies shorter training in exchange for the loss of information about the state of the environment. For this purpose, thresholds based on quartiles of the RSRP data collected in multiple experiments with different user distributions are used to create four classification levels. The cumulative distribution function (CDF) of the RSRP data from these simulations is represented in Figure 3. Moreover, the specific values of RSRP quartiles are detailed in Table I. Furthermore, it should be noted that an additional level is created to identify unused time slots. Hence, five levels are finally defined to achieve a balance between less training time and minimal loss of scenario information.

Finally, Algorithm 1 summarises the training of the DRL approach. Since the present study focuses on simulation analysis and no real tests are performed, the experiences of all robots are ideally collected to update the weights of a single DNN, thereby speeding up training. In this way, the simulation time decreases significantly. Once training is completed, the pre-trained DNN is deployed to all robots for distributed use.

IV. Evaluation

A. Simulation Assumptions

This subsection introduces further details about the scenario used to analyze the performance of the proposed methodology. Autonomous robots are randomly spawned in the indoor factory. They move at a constant speed between random locations. Regarding shared video streams in CCR, a data rate of 10

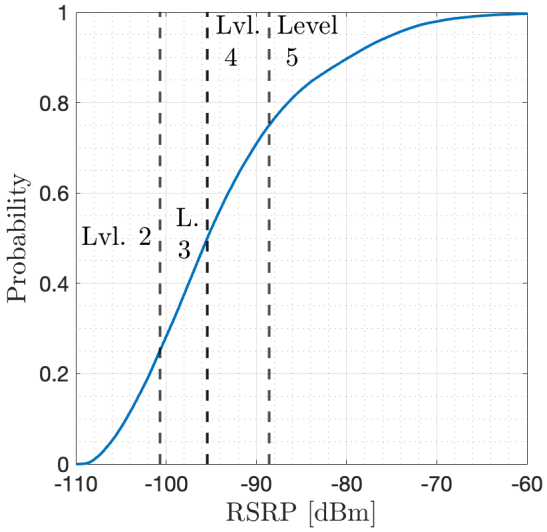

Fig. 3. CDF of the RSRP data collected in multiple experiments with different user distributions.

TABLE I
CLASSIFICATION LEVELS BASED ON THE RSRP QUARTILES

Level	RSRP
1	No device transmitting in this time slot
2	$RSRP \leq -100.7\ dBm$
3	$-100.7\ dBm < RSRP \leq -95.4\ dBm$
4	$-95.4\ dBm < RSRP \leq -88.6\ dBm$
5	$-88.6\ dBm < RSRP$

Mbps is targeted to be achieved [18]. The simulation parameters are summarized in Table II, e.g. the carrier frequency factor and reference offset which are used to calculate the path loss or the de-correlation distance which is used to generate the shadowing component. They are further detailed in [9].

Once the scenario is established, the robots are trained until the average reward per decision converges. Afterwards, the policy of the last training episode is used to assess the performance of the DRL methodology. This is then compared with the performance obtained by the NR sidelink mode 2 and a random resource allocation scheme, e.g. the Slotted ALOHA protocol [19]. This will allow to analyze the advantages of the DRL approach in terms of probability of successful reception for the intended 10 Mbps data rate. In this sense, the reception success depends on the signal-to-interference-plus-noise ratio (SINR) on reception and the corresponding block error rate (BLER) curves. Finally, several simulations are performed with a different number of robots in the scenario to analyze the behavior of the three schemes in environments with different device density.

B. Results and Discussion

Figure 4 shows the average reward per decision and the moving mean calculated over a 100 episodes window. The reward increases as training progresses. This increase becomes much slower once training exceeds 3000 episodes, and stops

Algorithm 1 Training of the DRL resource allocation scheme

Initialize θ arbitrary
for each episode **do**
  Initialize the location and trajectories of the robots randomly
  for each time step **do**
    All robots sense the resource pool
    for each robot with at least one other robot within CCR **do**
      $s_t \leftarrow$ Reduced and discretized RSRP frame sensed
      The robot inputs s_t into the DNN
      Selection of a_t based on the resulting probabilities of the softmax output unit
      Perform all the SPS transmissions
      $r_{t+1} \leftarrow$ (1)
      Store the experience (s_t, a_t, r_{t+1})
      if batch of T experiences is full **then**
        $G \leftarrow$ (2)
        $\Delta\theta \leftarrow$ (3)
        $\theta \leftarrow \theta + \Delta\theta$
        Store the total reward of the episode
        Reset the batch of T experiences
        Finish the episode
      end if
    end for
  end for
end for
Store the pre-trained DNN

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Carrier frequency	3.5 GHz
Critical cooperation range	5 m
Extended cooperation range	25 m
Facility dimensions	120 x 50 m^2 [10]
Transmission power	0 dBm
Bandwidth	100 MHz
NR slot duration	250 μs
Thermal noise power spectral density	-174 dBm/Hz
Receiver noise figure	9 dB
Interference	Independent intra-system interference
Device speed	1 m/s
Mobility model	Random waypoint (RWP)
Shadow fading standard deviation	5.7 [10]
Path loss coefficient	2.55 [10]
Carrier frequency factor	2 [10]
Reference offset	33 [10]
De-correlation distance	20 m [20]
Discovery message periodicity	100 ms
Data message periodicity	10 ms
Data message size	100 kb
Batch size, T	300
Discount factor, γ	0.2
Learning rate, α	0.0001

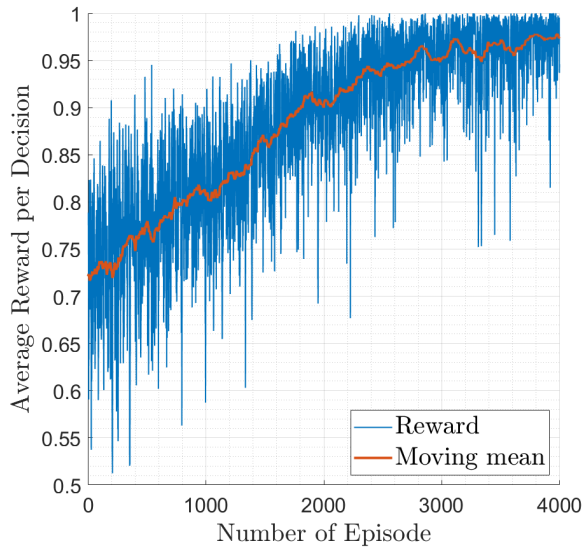

Fig. 4. Reward per action over the episodes of the training.

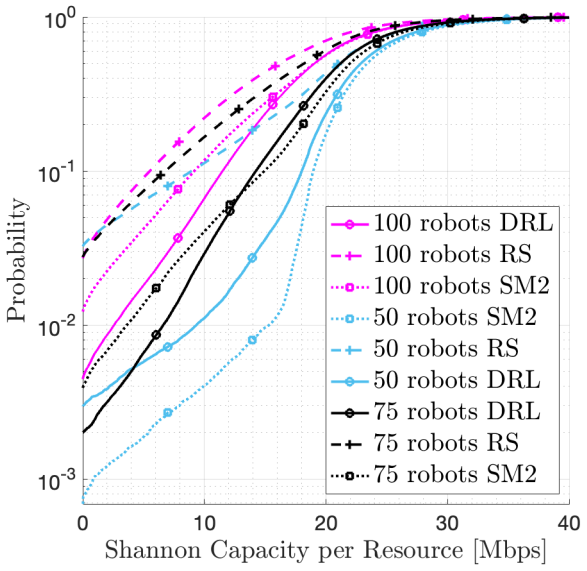

Fig. 5. CDF of per resource useful Shannon capacity for a different number of robots in the scenario.

increasing around 4000 episodes. Nonetheless, the average reward does not reach the maximum possible reward. Although the maximum reward is obtained in some episodes, the scenario will sometimes be crowded with nearby robots, leading to a fully utilized resource pool in which some collisions cannot be avoided. This also explains the variance obtained at the end of the training. It should be also noted that this variance decreases as the number of episodes increases, indicating that actions that provide the maximum possible reward are selected more frequently.

Next, the CDF of per resource useful Shannon capacity is calculated to check whether a 10 Mbps data rate is achieved by each of the introduced schemes in scenarios with different density. Figure 5 depicts the results for only three representative configurations for the sake of clarity. It shows that the NR sidelink mode 2 (SM2) achieves a better performance for a scenario with 50 robots than the DRL approach. This scenario configuration implies low slot occupancy. In contrast, the performance of the DRL approach is better in a very dense scenario with 100 robots. This is mainly because the selection phase of NR sidelink mode 2 removes slots already used by other robots from the set available for selection when the slot occupancy is low. As the number of devices increases, already occupied slots will become part of the remaining set of suitable slots. Hence, the slot random selection of the NR sidelink mode 2 last phase will not be effective in ensuring satisfactory reception in dense scenarios. On the other hand, the DRL scheme always decides among all slots regardless of whether other devices are transmitting in them. Moreover, the robots have previously acquired experience in dense situations. In this sense, some devices can reuse slots that are already occupied even if there are free slots, so that these free slots can be used by other devices with higher density of robots around them. For example, if two robots discover each other within the CCR and both detect a high percentage of occupied slots with similar but low RSRP, this would indicate that other areas of the factory away from these users have a high density of users. Therefore, these robots decide to reuse these slots so that potential new robots in areas equidistant between the introduced robots and the high density area can use the free slots. Otherwise, reuse of the slots by users closer to the dense zone could lead to collisions. Thus, some robots choose to transmit in slots that may involve a lower SINR but successful reception so that other robots in a more challenging situation have higher flexibility. However, this can also sometimes lead to unnecessary collisions in not too dense situations. This makes NR sidelink mode 2 work slightly better in case there is no need to reuse slots, and the DRL scheme performs better when the number of devices in the scenario is high and the same slots are often used by more than one device. Finally, it is also worth noting that the random scheme (RS) always achieves the worst performance.

Afterwards, the probability of successful reception is computed based on a 10 Mbps data rate. Figure 6 shows the complementary CDF (CCDF) of the probability of successful reception as the number of robots in the scenario increases. It shows that the performance of NR sidelink mode 2 achieves slightly better performance than the DRL scheme up to about 66 robots in the scenario. Thereafter, the improvement of the DRL approach over the NR sidelink mode 2 increases as the number of robots grows, reaching up to 5% higher probability of successful reception in a scenario with 100 users. Considering the performance of the analyzed approaches and the conclusions drawn, the use of NR sidelink mode 2 is proposed for scenarios where slots are not usually reused, due to its correct performance with a simpler deployment. However, if a larger number of devices are deployed, the DRL approach should be proposed to intelligently reuse time slots.

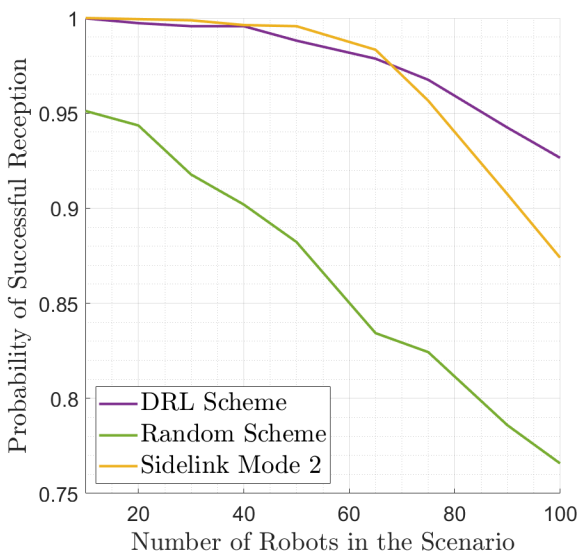

Fig. 6. CCDF of the probability of successful reception as the number of robots increases.

V. Conclusion and Further Work

The DRL methodology presented in this paper allows to significantly outperform the NR sidelink mode 2 approach in dense scenarios with high resource occupancy. In this regard, the proposed method provides the devices with the capability to better reuse slots already occupied by other devices. Furthermore, the proposed technique allows devices to manage radio resources autonomously based solely on their own view of the scenario, so no signalling exchange is required.

The performance of the methodology can be further improved via optimization of the DNN architecture and investigation on potential extension of the state space with other information available at the robots.

Acknowledgment

This work has been partially funded by Junta de Andalucía (projects EDEL4.0:UMA18-FEDERJA-172 and PENTA:PY18-4647) and Universidad de Málaga (I Plan Propio de Investigación, Transferencia y Divulgación Científica). Ramoni Adeogun is supported by the Danish Council for Independent Research, grant no. DFF 9041-00146B. The authors would like to express their profound gratitude to Nokia Standardization Aalborg and Aalborg University for funding the first author's research stay. The authors thank Assoc. Prof. Gilberto Beradinelli for his comments on the manuscript.

References

[1] I. O. Sanusi, K. M. Nasr and K. Moessner, "A Device to Device (D2D) Spectrum Sharing Scheme for Wireless Industrial Applications," 2019 European Conference on Networks and Communications (EuCNC), 2019, pp. 353-357.

[2] R. Adeogun, G. Berardinelli, I. Rodriguez and P. Mogensen, "Distributed Dynamic Channel Allocation in 6G in-X Subnetworks for Industrial Automation," 2020 IEEE Globecom Workshops (GC Wkshps), 2020.

[3] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm Robotic Behaviors and Current Applications," Frontiers in Robotics and AI, vol. 7, p. 36, Apr. 2020.

[4] Z. Yan and Y. Xu, "A Multi-Agent Deep Reinforcement Learning Method for Cooperative Load Frequency Control of a Multi-Area Power System," in IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4599-4608, Nov. 2020.

[5] S. Kim, D. Lee, I. Jang, H. Kim and Y. Son, "Periodic Communication for Distributed Multi-agent Reinforcement Learning under Partially Observable Environment," 2019 ICTC, Jeju, Korea (South), 2019, pp. 940-942.

[6] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," in Neurocomputing, vol. 190, pp. 82-94, May 2016.

[7] B. Gu, X. Zhang, Z. Lin and M. Alazab, "Deep Multiagent Reinforcement-Learning-Based Resource Allocation for Internet of Controllable Things," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3066-3074, 1 March, 2021.

[8] S. Yu, Y. J. Jeong and J. W. Lee, "Resource Allocation Scheme Based on Deep Reinforcement Learning for Device-to-Device Communications," 2021 International Conference on Information Networking (ICOIN), 2021, pp. 712-714.

[9] S. Morejon et al., "Cooperative Resource Allocation for Proximity Communication in Robotic Swarms in an Indoor Factory," 2021 IEEE Wireless Communications and Networking Conference (WCNC), 2021, pp. 1-6.

[10] 3rd Generation Partnership Project (3GPP), "Study on channel model for frequencies from 0.5 to 100 ghz," in 3GPP TR 38.901 V16.1.0, Dec. 2019.

[11] 3rd Generation Partnership Project (3GPP), "Physical layer procedures for data," in 3GPP TS 38.214 V16.5.0, March 2021.

[12] 3rd Generation Partnership Project (3GPP), "Medium Access Control (MAC) protocol specification," in 3GPP TS 38.321 V16.5.0, June 2021.

[13] S. Lien et al., "3GPP NR Sidelink Transmissions Toward 5G V2X," in IEEE Access, vol. 8, pp. 35368-35382, 2020.

[14] F. Cruz, P. Wüppen, A. Fazrie, C. Weber and S. Wermter, "Action Selection Methods in a Robotic Reinforcement Learning Scenario," 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Gudalajara, Mexico, 2018, pp. 1-6.

[15] A. Onat, N. Kosino, M. Kuramitu and H. Kita, "Reinforcement learning under incomplete perception using stochastic gradient ascent and recurrent neural networks," IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, 1999, pp. 481-486 vol.5.

[16] G. Chen, C. I. J. Douch and M. Zhang, "Using Learning Classifier Systems to Learn Stochastic Decision Policies," IEEE Transactions on Evolutionary Computation, vol. 19, no. 6, pp. 885-902, Dec. 2015.

[17] A. Kanervisto, C. Scheller and V. Hautamäki, "Action Space Shaping in Deep Reinforcement Learning," 2020 IEEE Conference on Games (CoG), Osaka, Japan, 2020, pp. 479-486.

[18] 3rd Generation Partnership Project (3GPP), "Study on enhancement of 3gpp support for 5g v2x services," in 3GPP TR 22.886 V16.2.0, Dec. 2018.

[19] K. Spathi, A. Valkanis, G. BeletsIoTi, G. Papadimitriou and P. Nicopolitidis, "Performance Evaluation of Slotted ALOHA based IoT Networks under Asymmetric Traffic," 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), 2020, pp. 1-5.

[20] S. Lu, J. May, and R. J. Haines, "Efficient modeling of correlated shadow fading in dense wireless multi-hop networks," in 2014 IEEE WCNC. Istanbul, Turkey: IEEE, Apr. 2014, pp. 311–316.