Implications of Decentralized Q-Learning Resource Allocation in Wireless Networks

Francesc Wilhelmi, Boris Bellalta
Wireless Networking (WN-UPF)
Univ. Pompeu Fabra
Barcelona

Cristina Cano
WINE Group
Univ. Oberta de Catalunya
Castelldefels, Barcelona

Anders Jonsson
Art. Int. and Mach. Learn. (AIML-UPF)
Univ. Pompeu Fabra
Barcelona

***Abstract*—Reinforcement Learning is gaining increasing attention by the wireless networking community due to its potential to learn good-performing configurations only from the observed results. In this work we propose a Stateless variation of Q-learning, which is applied to exploit spatial reuse in a wireless network. In particular, we allow nodes to modify both their transmission power and channel solely based on the experienced throughput. We concentrate in a completely decentralized scenario in which no information about neighbouring nodes is available to the learners. Our results show that although the algorithm is able to find the best-performing actions to enhance aggregate throughput, there is high variability in the throughput experienced by the individual networks. We identify the cause of this variability to be the adversarial setting of our setup in which the set of most played actions provide intermittent good/poor performance depending on the neighbouring decisions. The effect of the algorithm's learning parameters on this variability is also studied.**

I. Introduction

Reinforcement Learning (RL) has recently spread use in the wireless communications field to solve many kind of problems such as Access Point (AP) association, channel selection or transmit power adjustment [1, 2], as it allows learning good-performing configurations only from the observed results. Among these, Q-learning has been applied to dynamic channel assignment in mobile networks in [1] and to automatic channel selection in Femto Cell networks in [6]. However, the case of a fully decentralized scenario has not yet been considered.

In this work we propose a stateless variation of Q-learning in which nodes select the transmission power and channel to use solely based on the resulting throughput. We concentrate on a fully decentralized scenario where no information about the other nodes actions and resulting performance is available to the learners. Note that inferring the throughput of neighbouring nodes allocated to different channels is costly as periodic sensing in the other channels would then be needed. We aim to characterize the performance of Q-learning in such scenarios, obtaining insight on the most played actions (i.e., configurations selected) and the resulting performance. We observe that when no information about the neighbours is available to the learners, these will tend to apply selfish strategies that result in alternating good/poor performance depending on the actions of the others. In such scenarios, we show that the use of Q-learning allows each network to find the best-performing actions, though without reaching a steady solution. Note that achieving a steady solution in a decentralized environment relies in finding a Nash Equilibrium, a concept used in Game Theory to define a set of individual strategies that maximize the profits of each player in a non-cooperative game, regardless of the others' strategy. Formally, a set of best player actions $a^* = (a_1^*, ..., a_n^*) \in A$ leads to a Nash Equilibrium if $a_i^* \in B_i(a_{-i}^*), \forall i \in N$, where $B_i(a_{-i})$ is the best response to the others actions (a_{-i}). Thus, the consequences of not reaching a Nash Equilibrium can have an impact on performance variability.

In addition, we look at the resulting performance when varying several parameters intrinsic to the learning algorithm, which results to be helpful to understand the interactions among the degree of exploration and the variability of the resulting performance.

The remaining of this document is structured as follows: in Section II our Stateless variation of Q-learning and its practical implementation for the resource allocation problem in WNs is presented. Then, in Section III we present the simulation scenario and considerations. Simulation results are later discussed in Section IV. Finally, some final remarks are provided in Section V.

II. Decentralized Stateless Q-learning for enhancing Spatial Reuse in a WN

Q-learning [7, 8] is an RL technique that enables an agent to learn the optimal policy to follow in a given environment. A set of possible states (that describe the

status of the environment) and actions are defined. In particular, an agent maintains an estimate of the expected long-term discounted reward for each state-action pair, and selects actions with the aim of maximizing it. The expected cumulative reward $\mathrm{V}^{\pi}(s)$ is given by:

$$\mathrm{V}^{\pi}(s) = \lim_{N\to\infty} \mathbb{E}\left(\sum_{t=1}^{N} r_t^{\pi}(s)\right),$$

where $r_t^{\pi}(s)$ is the reward obtained at iteration t after starting from state s and by following policy π. Since the reward may easily get unbounded, a discount factor parameter ($\gamma < 1$) is used. The optimal policy π^* that maximizes the total expected reward is obtained by using the Bellman's Optimality Equation [7]:

$$Q^*(s,a) = \mathbb{E}\Big\{ r_{t+1} + \gamma \mathrm{max}_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a \Big\}.$$

Henceforth, Q-learning receives information about the current state-action tuple (s_t, a_t), the generated reward r_t and the next state s_{t+1}, in order to update the Q-table $\hat{Q}(s_t, a_t)$:

$$\hat{Q}(s_t,a_t) \leftarrow (1-\alpha_t)\hat{Q}(s_t,a_t) + \alpha_t(r_t + \gamma(\max_{a'}\hat{Q}(s_{t+1},a')),$$

where α_t is the learning rate at time t, and $\max_{a'}\hat{Q}(s_{t+1}, a')$ is the best estimated value for the next state s_{t+1}. The optimal solution is achieved with probability 1 if $\sum_{t=0}^{\infty}\alpha_t = \infty$, and $\sum_{t=0}^{\infty}\alpha_t^2 < \infty$, which satisfies that $\lim_{t\to\infty}\hat{Q}(s,a) = Q^*(s,a)$. Since we focus on a completely decentralized scenario where no information about the other nodes is available, the system can then be fully described by the set of actions and rewards.[1] Thus, we apply a Stateless variation of the original Q-learning algorithm. We then consider each WN to be an agent implementing Q-learning through an ε-greedy action-selection strategy, so that actions $a \in \mathcal{A}$ correspond to all the possible configurations that can be chosen with respect to the channel and transmit power. The implementation details are described in Algorithm 1.

Algorithm 1: Stateless Q-learning

Function Stateless Q-learning $(\mathrm{SINR}, \mathcal{A})$;
Input : SINR: Signal-to-Interference-plus-Noise Ratio sensed at the STA
$\mathcal{A}$: set of possible actions in {1, ..., K}
Output: $\overline{\Gamma}$: Mean throughput experienced in the WN
initialize: $t = 0$, $\hat{Q}(a_i) = 0, \forall a_i \in \mathcal{A}$
while *active* **do**
  Select a_i $\begin{cases} \underset{i=1,\ldots,K}{\text{argmax}}\ \hat{Q}(a_i), & \text{with prob } 1-\varepsilon \\ i \sim \mathcal{U}(1,K),, & \text{otherwise} \end{cases}$
  Observe reward $r_{a_i} = \frac{\Gamma_{a_i}}{\Gamma^*}$
  $\hat{Q}(a_i) \leftarrow \hat{Q}(a_i) + \alpha \cdot (r_{a_i} + \gamma \cdot \max(\hat{Q}(:)) - \hat{Q}(a_i))$
  $\varepsilon_t \leftarrow \varepsilon_0/\sqrt{t}$
  $t \leftarrow t+1$
end

III. System model

We consider a scenario at which several WNs are placed in a 3D-map (with parameters described later in Section IV-A), each one formed by an Access Point (AP) transmitting to a single Station (STA).

[1]We note that local information such as the observed instantaneous channel quality could be incorporated in the state definition. However, such a description of the system entails increased complexity.

A. Channel modelling

Path-loss and shadowing are modelled by the following log-distance model:

$$\mathrm{PL}_{i,j} = \mathrm{P}_{\mathrm{tx}_i} - \mathrm{P}_{\mathrm{rx}_j} = \\ = \mathrm{PL}_0 + 10 \cdot \alpha \cdot \log_{10}(d_{i,j}) + \mathrm{G_s} + \frac{d_{i,j}}{d_{\mathrm{obs}}}\mathrm{G_o},$$

where $\mathrm{P}_{\mathrm{tx}_i}$ is the transmitted power in dBm by WN w_i, $\mathrm{P}_{\mathrm{rx}_j}$ is the power in dBm received at WN w_j, PL_0 is the path-loss at 1 m in dB, d is the distance between the transmitter and the receiver in meters, $\mathrm{G_s}$ is the shadowing loss in dB, and $\mathrm{G_o}$ is the obstacles loss in dB. Note that we also include the factor d_{obs}, which represents the distance between two obstacles in meters.

B. Throughput calculation

By using the power received and the interference, we calculate the resulting throughput of each WN w_i by using the Shannon Capacity.

$$\Gamma_{i,t} = B \cdot \log_2(1 + \mathrm{SINR}_{\mathrm{w}_i,t}),$$

where B is the channel width and the experienced SINR is given by:

$$\mathrm{SINR}_{\mathrm{w}_i,t} = \frac{\mathrm{P}_{\mathrm{w}_i,t}}{\mathrm{I}_{\mathrm{w}_i,t} + \mathrm{N}}\ [dB], \qquad (1)$$

where $\mathrm{P}_{w_i,t}$ and $\mathrm{I}_{w_i,t}$ are the received power and the sum of the interference at WN w_i at time t, respectively, and N is the floor noise power. For each STA in a WN, the interference is considered to be the total power received

from all the APs of the other coexisting WNs $w_{j\neq i} \in \mathcal{W}$ as if they were continuously transmitting. Note that, adjacent channel interference is also considered in $\mathrm{I}_{w_i,t}$. We consider that the transmitted power leaked to adjacent channels is 20 dBm lower for each channel separation.

C. Reinforcement Learning Considerations

During the learning process we assume that WNs select actions sequentially, so that at each learning iteration, every agent takes an action in an ordered way. The order at which WNs choose an action at each iteration is randomly selected at the beginning of it. The reward after choosing an action is set as:

$$R_{\mathrm{w}_i,t} = \frac{\Gamma_{\mathrm{w}_i,t}}{\Gamma^*_{\mathrm{w}_i,t}},$$

where $\Gamma_{i,t}$ is the experienced throughput by w_i, and $\Gamma^*_{\mathrm{w}_i,t} = B{\cdot}\log_2(1{+}\mathrm{SNR}_{\mathrm{w}_i,t})$ is the maximum throughput at WN w_i (i.e., when it uses the maximum transmission power and there is no interference).

IV. Performance Evaluation

In this Section we introduce the simulation parameters and the considerations taken into account for the experiments.[2]. Then, we show the main results.

A. Simulation Parameters

According to [10], a typical high-density scenario for residential buildings contains $0.0033\mathrm{APs}/\mathrm{m}^3$. We then consider a map scenario with dimensions $10 \times 5 \times 10$ m containing 4 WNs that form a grid topology in which STAs are placed at the maximum possible distance from the other networks. This toy scenario allows us to study the performance of Stateless Q-learning in a controlled environment. We consider that the number of channels is equal to half the number of coexisting WNs, so that we can study a challenging situation regarding the spatial reuse. Table I details the parameters used.

B. Optimal solution

We first identify the optimal solutions that maximize: i) the aggregate throughput, and ii) the proportional fairness, which is computed as the logarithmic sum of the throughput experienced by each WN ($\sum_{\mathrm{w}_i\in\mathcal{W}} \log(\Gamma_{\mathrm{w}_i,t})$). The optimal solutions are listed in Table II. Note that, since the considered scenario is symmetric, there are two equivalent solutions. Note, as well, that in order to maximize the aggregate network throughput two of the WNs sacrifice themselves by choosing a lower transmit power. This result is then not likely to occur in an adversarial selfish setting.

[2]The code used for simulations can be found at https://github.com/wn-upf/Decentralized_Qlearning_Resource_Allocation_in_WNs.git.

Parameter	**Value**
Map size (m)	$10 \times 5 \times 10$
Number of coexistent WNs	4
APs/STAs per WN	1 / 1
Distance AP-STA (m)	$\sqrt{2}$
Number of Channels	2
Channel Bandwidth (MHz)	20
Initial channel selection model	Uniformly distributed
TPC Values (dBm)	{5, 10, 15, 20}
PL_0	5
G_s	Normally distributed with mean 9.5
G_o	Uniformly distributed with mean 30
f_w (meters to find a wall)	5
Noise level (dBm)	-100
Traffic model	Full buffer (downlink)

TABLE I: Simulation parameters

WN id	**Action that maximizes the Aggregate Throughput**	**Action that maximizes the Proportional Fairness**
1	1 (2)	7 (8)
2	1 (2)	8 (7)
3	7 (8)	7 (8)
4	8 (7)	8 (7)

TABLE II: Optimal configurations (action indexes) to achieve the maximum network throughput and prop. fairness, resulting in 1124 Mbps and 891 Mbps, respectively. In parenthesis the analogous solution is shown. Actions indexes range from 1 to 8 and are mapped to (channel number, transmit power (dBm)): {1,5}, {2,5}, {1,10}, {2,10}, {1,15}, {2,15},{1,20} and {2,20}, respectively.

C. Input Parameters Analysis

We first analyse the effects of modifying α (the learning rate), γ (the discount factor) and ϵ_0 (the initial exploration coefficient of the ϵ-greedy update rule) with respect to the achieved network throughput. We run simulations of 10000 iterations and capture the results of the last 5000 iterations to ensure that the initial transitory phase has ended. Each simulation is repeated 100 times.

Figure 1 shows the average aggregate throughput achieved for each of the proposed combinations. It can be observed that the best results with respect to the aggregate throughput, regarding both average and variance, are achieved when $\alpha = 1$, $\gamma = 0.95$ and $\varepsilon_0 = 1$. This means that for achieving the best results (i.e., high average aggregate throughput and low variance), the immediate reward of a given action must be considered rather than any previous information ($\alpha = 1$). We see that the difference between the pay-off offered by the best action and the current one must also be high ($\gamma = 0.95$). In addition, exploration must be highly

boosted at the beginning ($\epsilon_0 = 1$). For this setting, the resulting throughput (902.739 Mbps) represents 80.29% of the one provided by the optimal configuration that maximizes the aggregate throughput (shown in Table II). Regarding proportional fairness, the algorithm's resulting throughput is only 1.32% higher than the optimal (Table II). We also evaluate the relationship between different

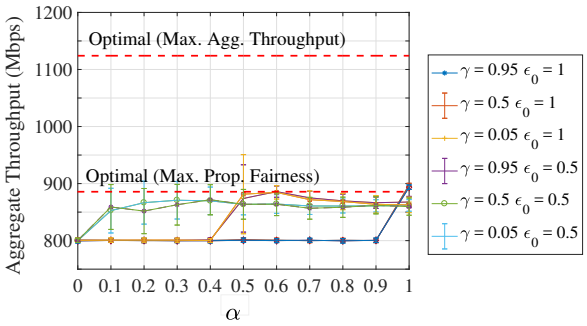
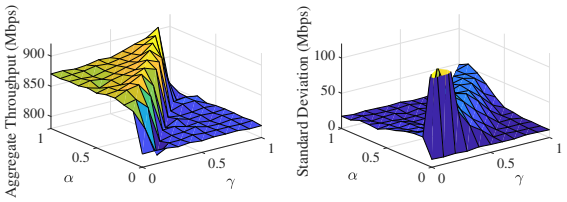


Fig. 1: Effect of α, γ and ε_0 in the average aggregate throughput (100 simulation runs per sample).

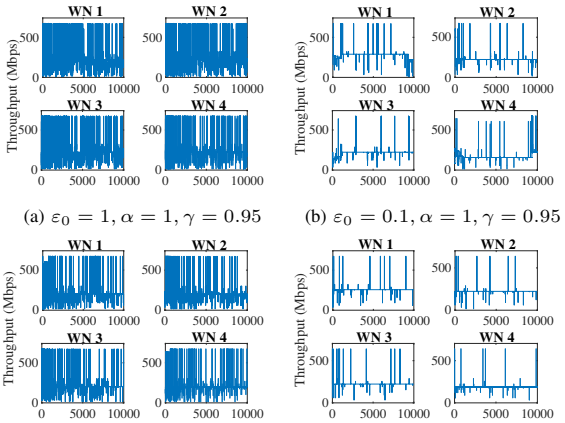
values of α and γ in the average aggregate throughput and standard deviation (shown in Figure 2). We can observe that we obtain consistently higher aggregate throughput when $\alpha > \gamma$. We also see that the variability between different simulation runs is much lower when the average throughput is higher. Additionally, we note a peak in the standard deviation when $\gamma \approx \alpha$ and $\gamma < \alpha$. To further understand the effects of modifying each of the aforementioned parameters, we show: i) the individual throughput experienced by each WN during the total 10000 iterations of a single simulation run (Figure 3), ii) the average throughput experienced by each WN for the last 5000 iterations, also for a single simulation run (Figure 4), and iii) the probability of choosing each action at each WN (Figure 5). As it can be observed (Figures 5(a), 5(c)), increasing the initial exploration coefficient (ε_0) allows for finding the best actions for each WN, so that the ones that provide consistently poor performance are discarded. However, the higher the exploration is, the higher the variability of the throughput experienced by each WN (Figures 3(a), 3(c)). By looking at the average throughput (Figure 4), we see that a higher exploration allows for fairer distribution of the channel resources at the expense of suffering higher throughput variability. When comparing Figures 5(a) and 5(c), we observe that for the former, there are two favourite actions that are being played the most, but for the latter there is only one preferred action. The lower the learning rate (α), and consequently



(a) Av. aggregate throughput (b) Standard deviation

Fig. 2: Evaluation of α and γ.

the discount factor (γ) due to the relationship shown in Figure 2, the higher the probability of choosing a unique action, which results to be the one that provided the best performance in the past. The opposite occurs for higher α and γ values, since giving more importance to the immediate reward allows for a reaction only to the recently-played actions of the neighbouring nodes: the algorithm is short-sighted.



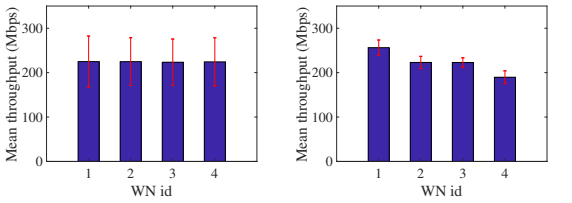
(a) $\varepsilon_0 = 1, \alpha = 1, \gamma = 0.95$ (b) $\varepsilon_0 = 0.1, \alpha = 1, \gamma = 0.95$

(c) $\varepsilon_0 = 1, \alpha = 0.1, \gamma = 0.05$ (d) $\varepsilon_0 = 0.1, \alpha = 0.1, \gamma = 0.05$

Fig. 3: Individual throughput experienced by each WN during a single (10000 iterations) simulation run for different ε_0, α and γ.

V. Conclusions

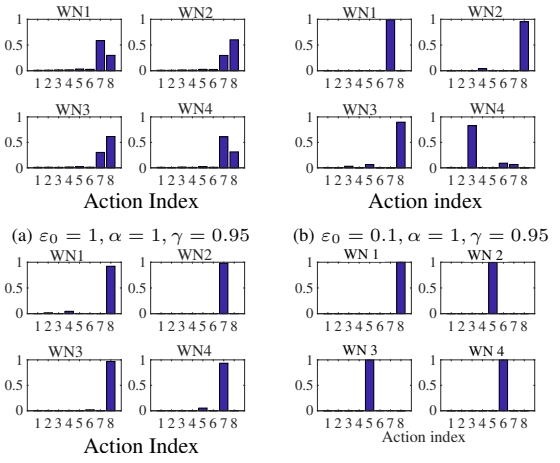
Decentralized Q-learning can be used to improve spatial reuse in dense wireless networks, enhancing performance as a result of exploiting the most rewarding actions. We have shown in this article, by means of a toy scenario, that Stateless Q-learning in particular allows finding good-performing configurations that achieve close-to-optimal (in terms of throughput maximization and proportional fairness) solutions. However, the competitiveness of the presented fully-decentralized



(a) $\varepsilon_0 = 1, \alpha = 1, \gamma = 0.95$ (b) $\varepsilon_0 = 0.1, \alpha = 1, \gamma = 0.95$

(c) $\varepsilon_0 = 1, \alpha = 0.1, \gamma = 0.05$ (d) $\varepsilon_0 = 0.1, \alpha = 0.1, \gamma = 0.05$

Fig. 4: Average throughput experienced by each WN during the last 5000 iterations of a total of 10000 iterations (in a single simulation run) and for different ε_0, α and γ.



(a) $\varepsilon_0 = 1, \alpha = 1, \gamma = 0.95$ (b) $\varepsilon_0 = 0.1, \alpha = 1, \gamma = 0.95$

(c) $\varepsilon_0 = 1, \alpha = 0.1, \gamma = 0.05$ (d) $\varepsilon_0 = 0.1, \alpha = 0.1, \gamma = 0.05$

Fig. 5: Probability of choosing the different actions at each WN for a single (10000 iterations) simulation run and different ε_0, α and γ values

environment involves the non-existence of a Nash Equilibrium. Thus, we have also identified high variability in the experienced individual throughput due to the constant changes of the played actions, motivated by the fact that the reward generated by each action changes according to the opponents' ones. We have evaluated the impact of the parameters intrinsic to the learning algorithm on this variability showing that it can be reduced by a decrease of the exploration degree and learning rate. This reduction on individual throughput variability occurs at the expense of aggregate performance. This variability can potentially result in negative effects in the overall WN's performance. The effects of such a fluctuation in higher layers of the protocol stack can have severe consequences depending on the time scale at which they occur. For example, high throughput fluctuations may trigger congestion recovery procedures in TCP (Transmission Control Protocol).

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), and by the European Regional Development Fund under grant TEC2015-71303-R (MINECO/FEDER).

REFERENCES

[1] Nie, J., & Haykin, S. (1999). A Q-learning-based dynamic channel assignment technique for mobile communication systems. IEEE Transactions on Vehicular Technology, 48(5), 1676-1687.

[2] Maghsudi, S., & Staczak, S. (2015). Joint channel selection and power control in infrastructureless wireless networks: A multiplayer multiarmed bandit framework. IEEE Transactions on Vehicular Technology, 64(10), 4565-4578.

[3] Akella, Aditya, et al. "Self-management in chaotic wireless deployments." Wireless Networks 13.6 (2007): 737-755.

[4] Riihijarvi, J., Petrova, M., & Mahonen, P. (2005, January). Frequency allocation for WLANs using graph colouring techniques. In Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on (pp. 216-222). IEEE.

[5] Mhatre, V. P., Papagiannaki, K., & Baccelli, F. (2007, May). Interference mitigation through power control in high density 802.11 WLANs. In INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE (pp. 535-543). IEEE.

[6] Bennis, M., & Niyato, D. (2010, December). A Q-learning based approach to interference avoidance in self-organized femtocell networks. In GLOBECOM Workshops (GC Wkshps), 2010 IEEE (pp. 706-710). IEEE.

[7] Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1, No. 1). Cambridge: MIT press.

[8] Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8 (3-4), 279-292.

[9] Jain, R., Durresi, A., & Babic, G. (1999). Throughput fairness index: An explanation (pp. 99-0045). Tech. rep., Department of CIS, The Ohio State University.

[10] Bellalta, Boris. "IEEE 802.11 ax: High-efficiency WLANs." IEEE Wireless Communications 23.1 (2016): 38-46.