Asynchronous vs Synchronous Selfish Learning

October 10, 2018

1 Introduction

In order to decrease the temporal variability experienced by WNs applying online learning, we propose the following mechanism:

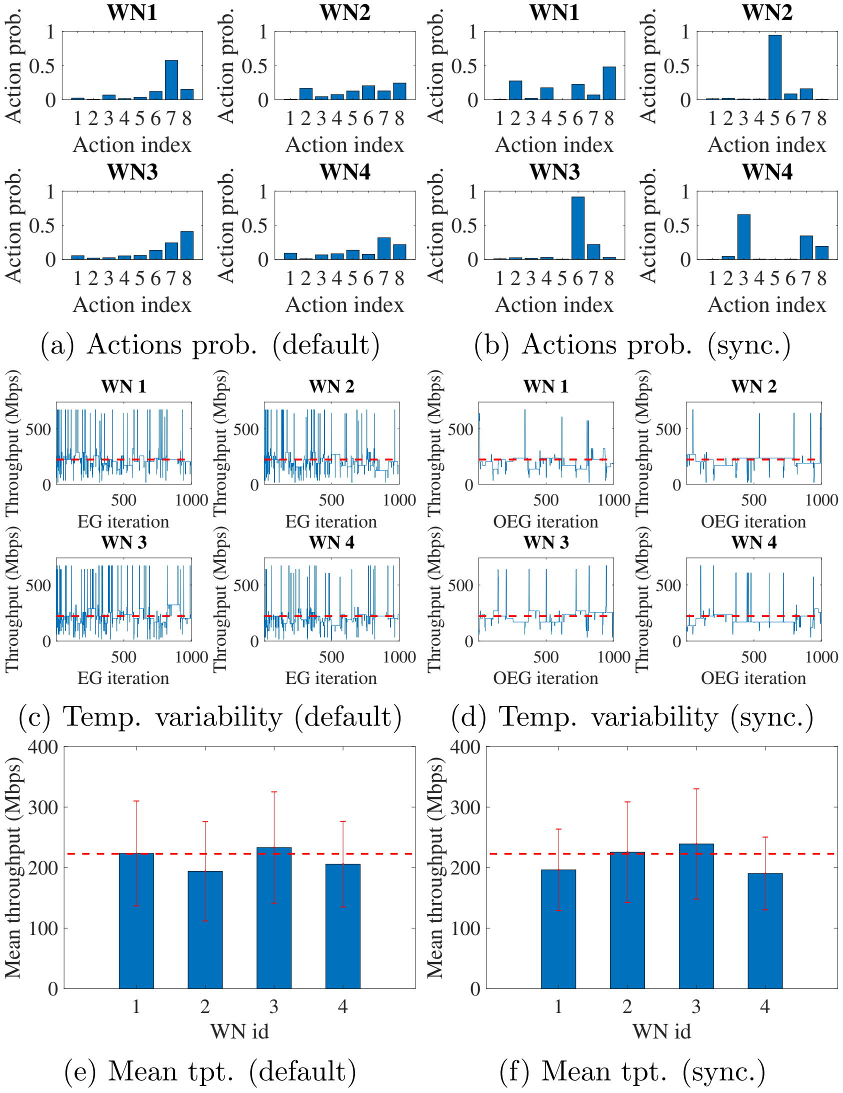
- WNs are synchronized, so that they act at specific intervals.
- In each iteration, only one WN is able to select an action. Of course, it can select the one that is currently using (exploitation).
- Every WN that remains "static" in a given iteration is considered to choose the last selected action.

Two important implications are derived from the abovementioned mechanism: *i)* the action-selection strategy of the learning algorithms are modified, *ii)* the adversarial setting changes, so that the performance of every chosen action is potentially evaluated in a larger set of situations. According to the latter, a higher level of valuable knowledge is more likely to be provided to each arm. However, depending on the learning algorithm used, there can be other implications.

2 Results - Toy Scenario

2.1 ε-greedy

Figure 1 shows several results regarding the application of ε-greedy in the toy scenario (1,000 iterations are considered), both for the fully decentralized and the synchronized approaches. As shown, the latter allows to experience a lower temporal throughput variability (1(c) vs 1(d)), since a lower number of actions is being exploited (1(a) vs 1(b)). However, similar results are obtained with respect to the average throughput experienced per WN (1(e) vs 1(f)). The fact is that, for the synchronized approach, WNs tend to exploit sub-optimal actions for longer periods (while they wait for their turn). In contrast, WNs are able to rapidly discard these sub-optimal actions in their turn.



(a) Actions prob. (default) (b) Actions prob. (sync.)

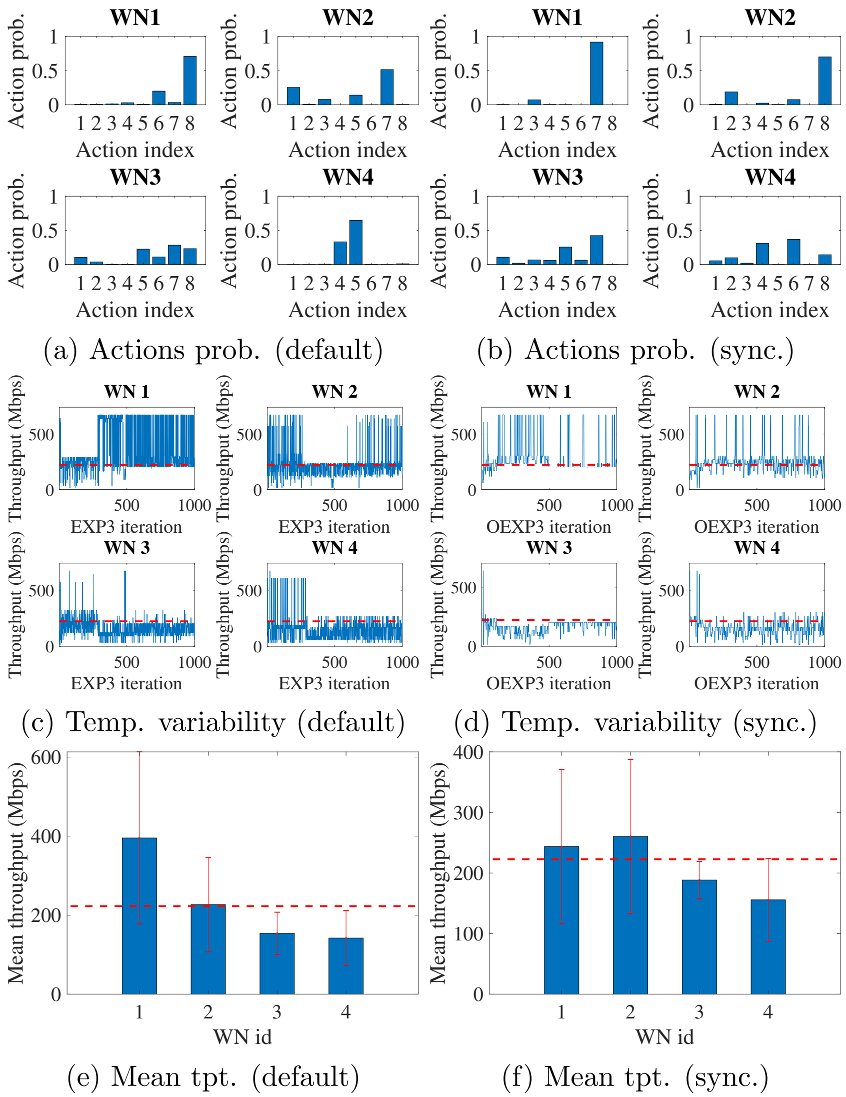
(c) Temp. variability (default) (d) Temp. variability (sync.)

(e) Mean tpt. (default) (f) Mean tpt. (sync.)

Figure 1: Simulation results in ε-greedy (1,000 iterations)

2.2 EXP3

Figure 2 shows several results regarding the application of EXP3 in the toy scenario (1,000 iterations are considered), both for the fully decentralized and the synchronized approaches. Again, the synchronized version of EXP3 leads to a lower temporal variability (2(c) vs 2(d)). No significant conclusions can be drawn for the actions probability profile (2(a) vs 2(b)), since in EXP3 it appears to be slightly random (recall that different EXP3 simulations may lead to different results). Finally, regarding the average throughput experienced per WN, similar "pseudo-randomized" results are obtained (2(e) vs 2(f)).



(a) Actions prob. (default) (b) Actions prob. (sync.)

(c) Temp. variability (default) (d) Temp. variability (sync.)

(e) Mean tpt. (default) (f) Mean tpt. (sync.)

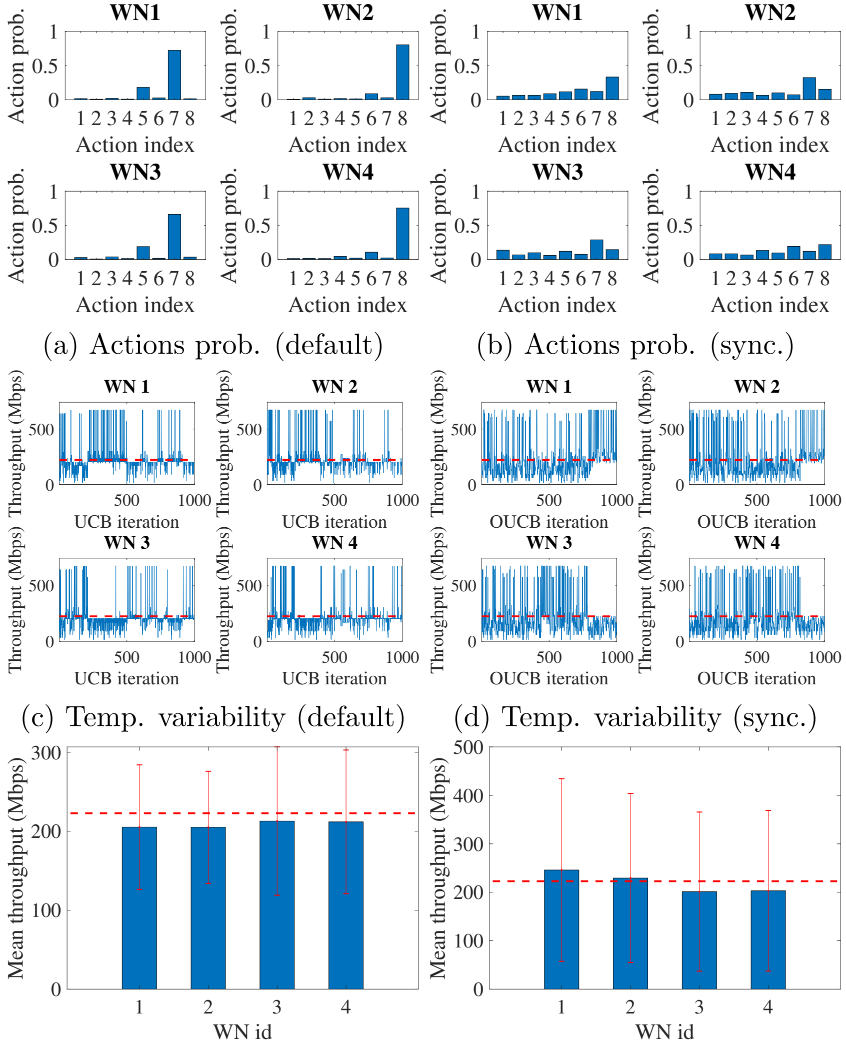
Figure 2: Simulation results in EXP3 (1,000 iterations)

2.3 UCB

Figure 3 shows several results regarding the application of UCB in the toy scenario (1,000 iterations are considered), both for the fully decentralized and the synchronized approaches. Unlike the previous cases, applying the synchronized approach to UCB is counter-productive, since its operational mode is altered by forcing WNs to choose undesired actions. As a result, the action-selection strategy is somehow randomized.

2.4 Thompson sampling

Figure 4 shows several results regarding the application of Thompson sampling in the toy scenario (1,000 iterations are considered), both for the fully decentralized and the synchro-



(a) Actions prob. (default)

(b) Actions prob. (sync.)

(c) Temp. variability (default)

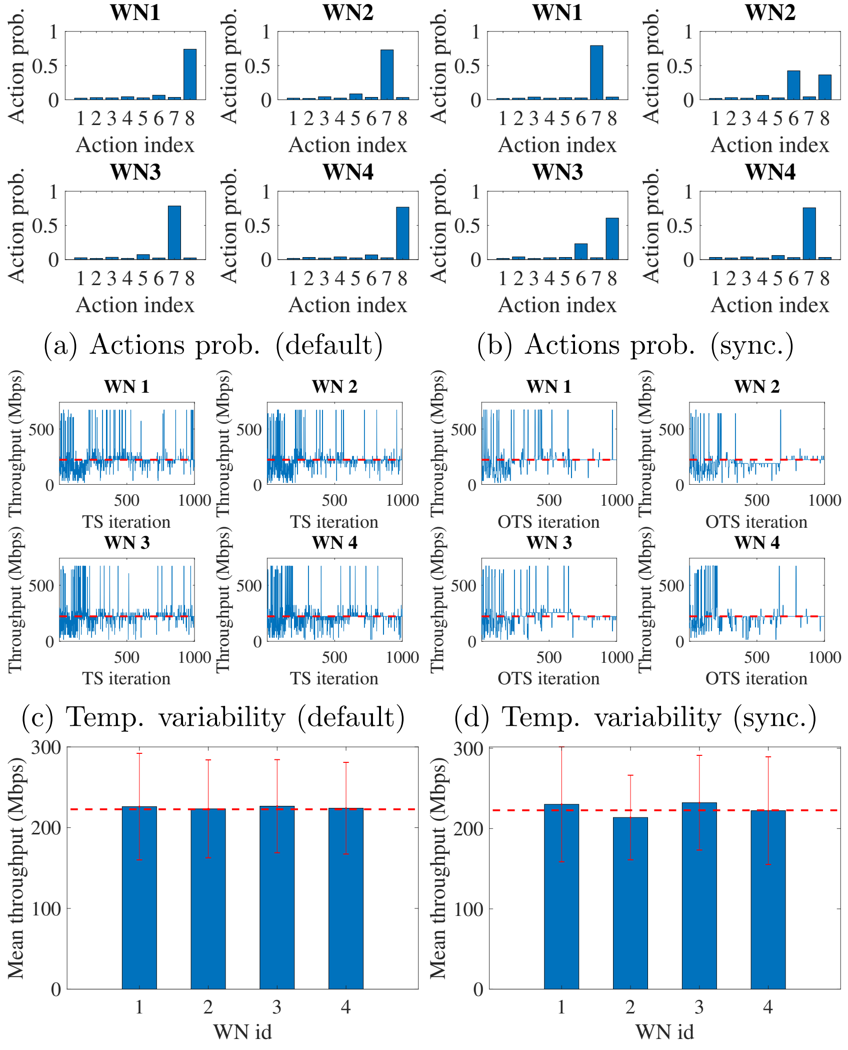
(d) Temp. variability (sync.)

(e) Mean tpt. (default)

(f) Mean tpt. (sync.)

Figure 3: Simulation results in UCB (1,000 iterations)

nized approaches. In this case, the temporal variability is significantly decreased (4(c) vs 4(d)). In contrast, the actions probability changes for the synchronized approach. In this case, some WNs (2 and 3) alternate two between actions. As it can be inferred from the temporal variability, one of the actions (the one granting lower performance) is selected until iteration 600 (approximately). At that moment, the best performing action (with respect to the adversarial environment) is chosen. This phenomena allegedly occurs when near-to-optimal actions are forced to be selected several times during the ”static” periods. As a result, the algorithm provides high estimates to that actions, which are hard to be overtaken by the actual best-performing ones.



(a) Actions prob. (default) (b) Actions prob. (sync.)

(c) Temp. variability (default) (d) Temp. variability (sync.)

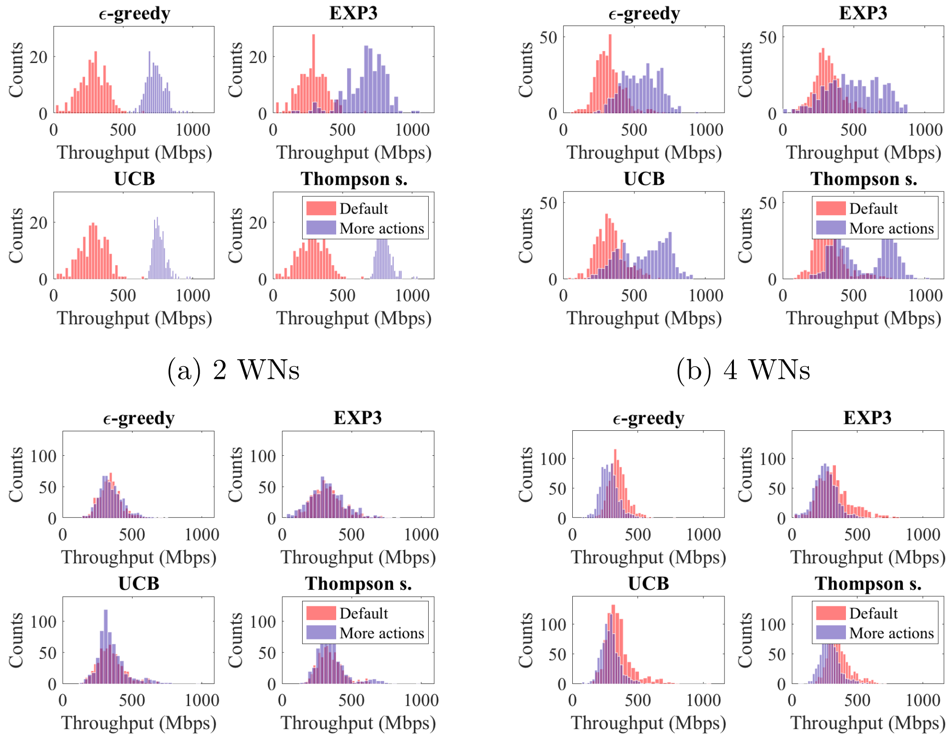
(e) Mean tpt. (default) (f) Mean tpt. (sync.)

Figure 4: Simulation results in Thompson sampling (1,000 iterations)

3 Results - Random Scenario

In order to reduce the simulation time, we have considered 1,000 learning iterations, rather than 100,000.

First, we aim to show the effect of considering a higher number of actions, thus framing more realistic settings. For that, we have considered to use 3 orthogonal channels (as for the 2.4 GHz band), and 4 levels of power, which now range from -15 to 30 dBm (as defined in the IEEE 802.11 standard). As shown in Figure 5, using the new set of actions leads, in general, to much better and concentrated results for low densities (5(a)). However, as the number of WNs increases, the performance achieved by the new set of actions is worst than the achieved by the old set. Recall that the old set of actions considers that the number of orthogonal channels increases according to density (i.e., $n_{channels} = \frac{n_{WNs}}{2}$). Finally, it is interesting to observe the case where we have 4 WNs (5(b)). In this case, there are two clearly separated sets of throughputs when using the new set of actions, which is more evident for UCB and Thompson sampling. The fact is that there are 3 orthogonal channels and 4 WNs. Therefore, with a proper channel allocation, two WNs will enjoy full channel access, while the other two will compete for the remaining one.



(a) 2 WNs

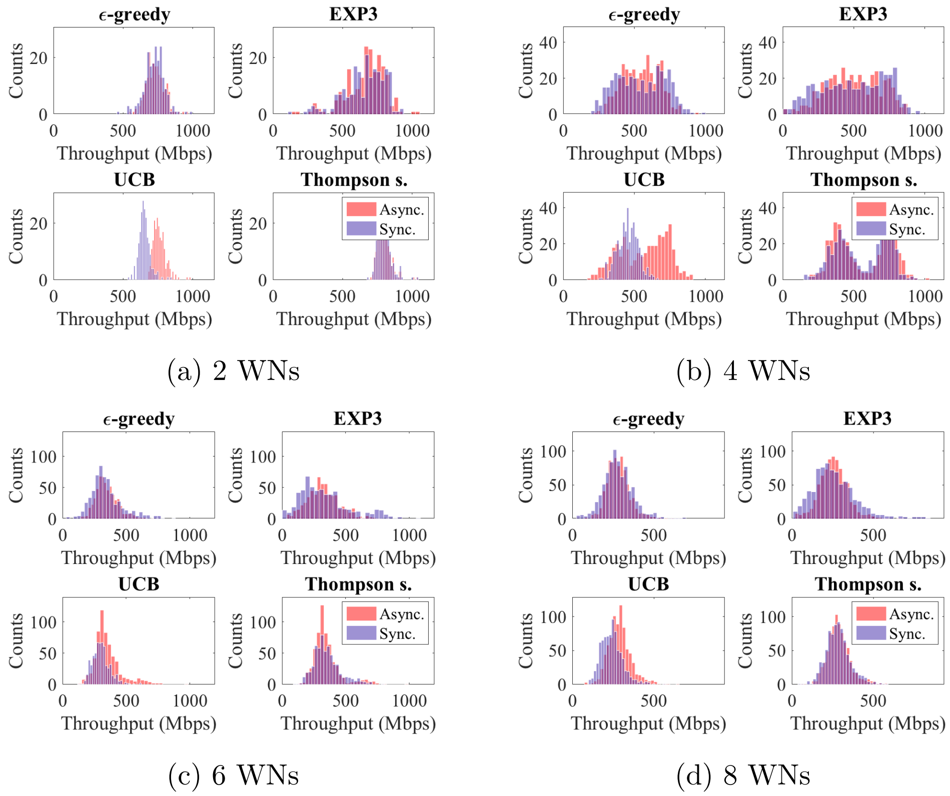
(b) 4 WNs

(c) 6 WNs

(d) 8 WNs

Figure 5: Histogram of the mean throughput experienced by WNs for each algorithm. The results are computed from 100 different random scenarios for each of the WNs sizes, where 1,000 learning iterations are considered. In red, we have the results for the initial set of actions, which consider that the number of channels is half the number of WNs. In blue, the new set of actions is considered.

Now, in order to further illustrate the effect of applying synchronized selfish learning, we compare it with the default decentralized approach in the random scenarios. In this case, we use the realistic range of actions in both situations. As shown in Figure 6, the synchronized mechanisms lead to less variability than their default versions (the histogram tends to be narrower for all the cases). In contrast, some performance loss can be observed, especially for the UCB algorithm (note that the blue distribution is slightly shifted to the left, with respect to the red one).



(a) 2 WNs

(b) 4 WNs

(c) 6 WNs

(d) 8 WNs

Figure 6: Histogram of the mean throughput experienced by WNs for each algorithm. The results are computed from 100 different random scenarios for each of the WNs sizes, where 1,000 learning iterations are considered. In red, we have the results for the default algorithms implementation, while blue indicates their synchronized versions.

4 Impact of modifying the range of actions

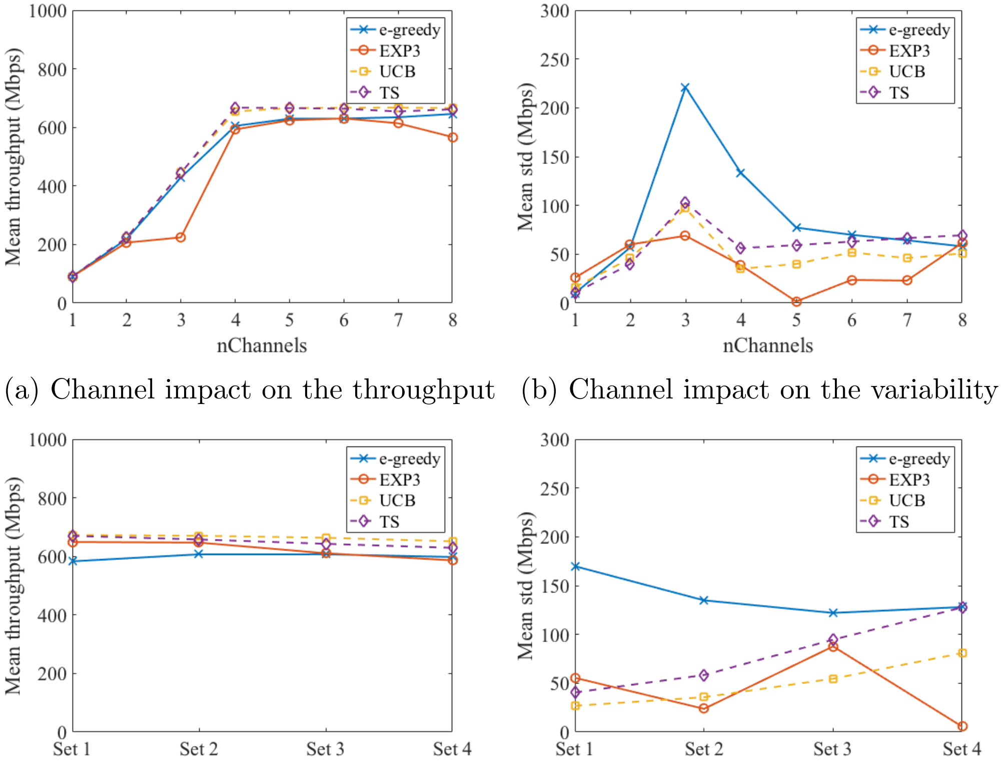
Here we aim to measure the impact of modifying the range of actions on the performance of the learning algorithms. For that, for each of the proposed action-selection strategies, we will compare the following performance measurements in the grid scenario, for different number of available actions: *i)* the mean throughput experienced in average, *ii)* the mean variability experienced in average.

Firstly, Figures 7(a) and 7(b) show the impact of using different number of orthogonal channels for each algorithm. As shown in 7(a), the mean throughput increases until reaching 4 channels, so that each WN is able to properly access to a single channel. Such an ideal case is achieved by Thompson sampling and UCB. However, for ε-greedy and EXP3 we notice some performance anomalies when the number of available channels increases. Regarding the experienced variability (7(b)), we observe a peak when the number of channels is 3, situation in which at least two WNs are permanently in conflict, since they are competing for the same channel resources. Note that, for the other cases where the number of channels is not enough to accommodate all the WNs, a lower variability is experienced due to the impossibility of achieving a better performance.

Secondly, the impact of modifying the available levels of transmit power is shown in both 7(c) and 7(d). In this case, we use the following sets of transmit power levels:

- Set 1: {5, 20} dBm
- Set 2: {5, 10, 15, 20} dBm
- Set 3: from 2 to 20 dBm in steps of 2 dBm
- Set 4: from 1 to 20 dBm in steps of 1 dBm

Regarding the throughput experienced (7(c)), we notice a very low variability when different sets of transmit power levels are used. In contrast, increasing the number of transmission power levels affects to the variability experienced by WNs. Such a variability increases in case of UCB and Thompson sampling. Different effects are noticed for ε-greedy and EXP3.



(a) Channel impact on the throughput (b) Channel impact on the variability

(c) TxPow impact on the throughput (d) TxPow impact on the variability

Figure 7: Mean and standard deviation of the throughput experienced in average when using different sets of actions.