Reinforcement learning approach for Advanced Sleep Modes management in 5G networks

Fatma Ezzahra Salem[a,b], Zwi Altman[a], Azeddine Gati[a], Tijani Chahed[b], Eitan Altman[c]
[a]Orange Labs, Châtillon, France
[b]Institut Mines-Telecom, Telecom SudParis, UMR CNRS SAMOVAR, Evry, France
[c]INRIA, Sophia-Antipolis, France
Email: {fatma.salem,azeddine.gati,zwi.altman}@orange.com
tijani.chahed@telecom-sudparis.eu eitan.altman@inria.fr

***Abstract*—Advanced Sleep Modes (ASMs) correspond to a gradual deactivation of the Base Station (BS)'s components in order to reduce its Energy Consumption (EC). Different levels of Sleep Modes (SMs) can be considered according to the transition time (deactivation and activation durations) of each component. We propose in this paper a management solution for ASMs based on Q-learning approach. The target is to find the optimal durations for each SM level according to the requirements of the network operator in terms of EC reduction and delay constraints. The proposed solution shows that even with a high constraint on the delay, we can achieve high energy savings (almost 57% of EC reduction) without inducing any impact on the delay. When the delay constraint is relaxed, we can achieve up to almost 90% of energy savings.**

***Index Terms*—Advanced Sleep Modes, Energy Consumption, Q-learning.**

I. Introduction

Motivated by both economic and environmental concerns, Energy Consumption (EC) in cellular networks has sparked wide interest in academia and industry during the last decade and has become one of the key pillars in the design of future 5G networks [1]. On the one hand, reducing the EC consumption of the networks enables to reduce the carbon emissions in the atmosphere, knowing that Information and Communication Technologies (ICT) systems are responsible for 2% of the world's CO_2 emissions [2]. On the other hand, the EC reduction will lead to lower Operational Expenditures (OPEX) for the network operators. These motivations led to the creation of a research direction named "Green Radio" dedicated to solutions that enable to build future wireless architectures ensuring better coverage and enhanced Energy Efficiency (EE) [3], defined as the amount of energy transmitted per Joule of consumed energy [4].

Since the Base Stations (BS) are the main energy consumers in the wireless network with around 80% of the total EC [5], many research studies focused on finding effective solutions to enhance their EE. In this context, some works considered reducing the BS transmit power, others focused on the hardware efficiency while many others considered the opportunity of deactivating the BSs, i.e, putting them into Sleep Mode (SM), whenever it is possible [6].

We focus in this work on the technique of SMs, and more precisely on a feature called Advanced Sleep Modes (ASMs). It corresponds to a gradual deactivation of the BS's components according to the time each of them needs to deactivate then reactivate again (transition time). Based on this time, we can define different levels of SMs having different characteristics, such as duration and power consumption [7]. It has been shown in a previous work [8] that this technique is very efficient in terms of EC since it can reduce up to 90% at very low loads where the ASMs are more applicable.

However, this technique induces an increase of the latency due to the waiting time of a user requesting a service while the BS is in SM. Therefore, a smart management solution is needed in order to find the optimal tradeoff between EC and delay. This solution should enable the network operator to orchestrate the ASMs according to its needs and to the different requirements of 5G use cases. For instance, if we are facing a delay-critical use case, such as Utra-reliable Low Latency Communications (URLLC) in 5G [9], there is a high constraint on latency which has to be in the order of 1ms. In this case, we should avoid ASMs inducing large delays. In other delay-tolerant use cases, the network operator has the option to prioritize either the energy reduction or the delay.

The main objective is to build a self tuned network which can adapt the use of the different ASM levels according to a cost function chosen by the network operator. This energy-saving mechanism can be viewed as a Green Self-organized Networking (SON) [10] function to be integrated into the network in addition to the other existing SON functions, such as those ensuring coverage and mobility.

In this paper we propose a Reinforcement Learning (RL) solution with a Q-Learning implementation in order to derive a controller that efficiently activates ASMs according to the desired utility. We focus on a distributed architecture where each BS uses solely its local information in order to learn the energy-saving policy. To the best of our knowledge, this is a first attempt to define an intelligent control system enabling to choose between different SM levels having different power values and different transition times.

The remainder of this paper is as follows: Section II introduces the concept of ASMs and describes the implementation

proposal. Section III presents our Q-learning approach adapted to the ASM problem. Section IV presents the numerical results and Section V concludes the paper.

II. System description

A. Advanced Sleep Modes

ASMs correspond to an energy-efficient feature which consists in deactivating the different components of the BS gradually. Therefore, different types of sleep levels can be considered according to the transition time of each component, i.e., the time needed to shut down the component then wake it up again. For instance, the Power Amplifier (PA) needs only one OFDM symbol (71μs) for this transition time while other components like the digital baseband need more time [7]. This led to the categorisation of the different components of the BS. Each category (SM level) comprises the components having the same transition time. Going from one level of SM to a deeper one allows more energy reduction since we deactivate more hardware; the BS needs however more time to reactivate them to serve the users. We consider four levels of SMs as introduced in [7]. Table II summarizes their different characteristics:

TABLE I: Advanced Sleep Modes characteristics

Sleep level	Deactivation duration	Minimum sleep duration	Activation duration
SM$_1$	35.5 μs	71 μs	35.5 μs
SM$_2$	0.5 ms	1 ms	0.5 ms
SM$_3$	5 ms	10 ms	5 ms
SM$_4$	0.5 s	1 s	0.5 s

Moreover, the BS has to send periodically signaling bursts. It has been agreed in 3GPP [11] that this peridicity can be configurable in 5G networks and can be set to any value among [5, 10, 20, 40, 80, 160 ms]. With these values of signaling periodicities, the deepest SM, i.e., SM$_4$, cannot be used. Then, we limit our interest in this work to the first three levels.

B. ASM implementation proposal

We proposed in [8] to implement the ASM in a gradual fashion. In other words, if the BS is totally idle (not serving any user) we put it into SM$_1$, then SM$_2$ and finally SM$_3$ as shown in Figure 1.

Whenever a user requests a service while the BS is in SM, we buffer it and trigger the activation of the BS. This can induce high impact on the latency since the considered user has to wait while the BS reactivates. This waiting time can reach 5ms if the BS is in SM$_3$, which can be very frequent in very low loads if we use the gradual approach described above. This may be critical if we consider delay-intolerant 5G use cases such as URLLC. So the challenge in this case is to find other solutions enabling to reduce the EC as much as possible using the ASMs without inducing any impact on the latency. For other use cases, for instance enhanced Mobile BroadBand (eMBB), the network operator can define another

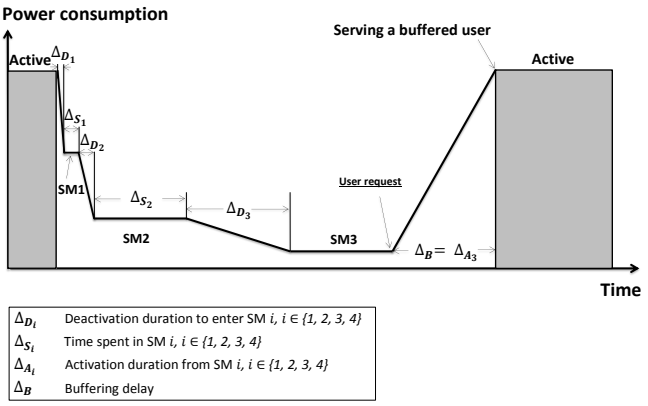

Fig. 1: Example of implementation strategy of ASMs

policy, such as choosing to reduce the EC without having any constraint on the delay. Therefore, our target is to design a network orchestrator enbaling to manage these ASMs in an optimal manner satisfying the network operator's needs.

In order to be able to save energy without inducing any delay, we propose the following implementation strategy: after a departure of a user and if the BS becomes totally idle, we put it into the most energy saving SM level, i.e., SM$_3$. We have to choose the number of times we can repeat SM$_3$ before going to the next SM level. After the time allowed for SM$_3$ elapses, we switch the BS into SM$_2$ and similarly the number of times SM$_2$ will be repeated has to be fixed at the end of SM$_3$. After SM$_2$ is finished, we have to decide how long we can stay in SM$_1$ before waking up totally. If there is a high constraint on the delay, our orchestrator has to anticipate the wake up of the BS, assuming that it will have less energy savings. Whereas, if the only constraint is on the EC, the orchestrator would prefer to stay in the depeest SM.

The following section introduces the machine learning approach that we use to solve this problem.

III. Q-learning proposal for dynamic orchestration of ASMs

A. Reinforcement learning approach

RL is a machine learning approach that aims to achieve an optimal goal by interacting with an environment. It consists on learning how to map situations to actions in order to maximize a numerical outcome [12]. This outcome can have a negative value, in this case it is considered as a cost or penalty and it can be positive, it is seen in this case as a reward. It indicates to the decision maker entity, called also *agent* or *controller*, whether the actions he chose are appropriate for the environment or should be avoided in the future. The environment is defined as everything being exterior to the agent. As the environment evolves in time, the agent has to adapt itself and to learn it continously.

By interacting with the environment, the agent acquires some knowledge which he can *exploit* when making his future

decisions. He can also choose to *explore* new actions in order to discover the optimal solution and be able to achieve better rewards in the future. Choosing the exploitation approach is the best thing to do so as to maximize the reward in the following step but the exploration has the perk of producing the best global reward in the long run. Hence, the optimal approach is to find a good tradeoff between the exploration and the exploitation strategies. One common solution to tackle this problem is the ϵ-greedy algorithm defined as follows:

$$Next\ action = \begin{cases} A^* & \text{with probability 1 - } \epsilon_{exp} \\ Random\ action & \text{with probability } \epsilon_{exp} \end{cases} \quad (1)$$

with A^* the best action known so far and $\epsilon_{exp} \in [0, 1]$ defines the probability of exploration.

Several methods exist in the literature that enable to solve a RL problem. They can be classified into two categories: model-based approaches such as Dynamic Programming and Monte Carlo and model-free approaches such as Temporal-Difference (TD) methods [12]. Two common examples of TD techniques are SARSA and Q-learning [13].

The following section describes our approach for ASM management based on Q-learning.

B. Q-learning proposal

As an application to the ASM problem, TD is a suitable approach since we do not have an explicit model of our environment (for example the transition probabilities between the SMs levels are not known). In this work, we use the Q-learning algorithm which is a control method where the agent can behave randomly without any specific policy.

We consider that the decisions are taken whenever the BS switches from an active mode to an idle one and whenever the sleep period for a certain SM level elapses. Let n_i denote the number of times the BS can repeat SM_i, for $i \in \{1, 2, 3\}$. We define an episode as the time between the departure of the last user served by the BS and the arrival of the next one. The beginning of an episode represents the decision point for the agent to choose n_3, the number of times we can repeat SM_3. Similarly, after spending n_3 times in SM_3, the BS has to take a decision for the next step: choose n_2 and the same for n_1 to be chosen after the time allowed of SM_2 elapses. So, the actions are taken at transition points between SM levels. The type of transition is defined by the policy, namely (n_3, n_2, n_1), and the actions consist of the number of times the next SM is repeated. Note that according to the policy defined above, if the system is in a SM_i, it is repeated a number of times n_i. Hence, the decision times for this problem are not fixed but they are flexible depending on the users' and SMs' dynamics in the network. Let us denote by t_0 the time of departure of the last user served by the BS. So the set of decision times for our problem can be written as follows:

$$T = \{t_0, t_0+n_3T_3, t_0+n_3T_3+n_2T_2, t_0+n_3T_3+n_2T_2+n_1T_1\}.$$

where T_i denotes the minimum duration for SM_i. At time $t_0 + n_3T_3 + n_2T_2 + n_1T_1$, the only possible action for the BS is to wake up. It corresponds to the terminal state in which the BS will remain until the arrival of the next user. The departure of that user, if no one else is being served by the BS by that time, corresponds to the start of the next episode. If a user arrives while the BS is in SM, the activation of the BS is triggered and the user has to wait.

The state space S comprises the states of the BS, i.e., active (serving a user), in sleep mode (SM_1, SM_2 or SM_3), or idle (not transmitting anything but it is still activated).

$$S = \{\text{active, idle, } SM_1, SM_2, SM_3\}.$$

The action space contains the possible decisions to be made at each decision point. The action is to choose how many times the BS can stay in the following SM level. We denote N_i the set of possible values that n_i can take, for $i \in \{1, 2, 3\}$.
So, the action space is:

$$A_s = \{N_3, N_2, N_1\}.$$

At each decision point, the agent chooses an action then stores a quality-value linking the state $s \in S$ to the chosen action $a \in A_s$. This quality-value, called Q-value, is initially taken as zero. The actions of the agent consist of selecting the number of times to repeat a SM level. This choice is based on the exploration-exploitation algorithm presented in Eq. (1). This means that the agent will choose a SM level having the highest Q-value with a probability 1 - ϵ_{exp} and a random level with probability ϵ_{exp}. Whenever the agent visits the state s and performs action a, it receives a reward R and the corresponding Q-value, $Q(s, a)$ is updated following this update-rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma max'_a Q(s', a') - Q(s,a)] \quad (2)$$

where:

- α is the learning rate. It shows how the algorithm will adapt to a new reward value.
- γ is the discount factor. It reflects the weight given to future rewards.
- s' is the state of the system after having performed action a in state s.

Our target is to find the best policy, i.e., the optimal triplet (n_3, n_2, n_1), which maximizes the reward R. We define R for a given episode as the weighted sum of the energy gain E_{gain} and the added delay D, both resulting from the sleep strategy during that episode. Hence, R can be written as follows:

$$R = -\epsilon D + (1 - \epsilon)E_{gain} \quad (3)$$

where ϵ is a normalized weight ($\epsilon \in [0, 1]$) that denotes the importance given to the two factors D and E_{gain}. A small ϵ means that the EC reduction is prioritized over the delay and vice versa.

IV. Numerical analysis

A. Simulator description

To evaluate the performance of the proposed approach, we developped an event-based simulator where an event corresponds to an arrival or a departure of a user in the network. We consider a single BS with one sector and a FTP service where the users request to download a file of exponential size with mean 4 Mbits. The end of the service correspond to their departure from the BS. The different characteristics of the simulator are given in Table II.

TABLE II: Simulator characteristics

Network parameters	
Antenna height	30 m
Bandwidth	20MHz
Scheduling type	Round Robin
Channel characteristics	
Thermal noise	-174 dBm/Hz
Path loss (d in km)	128.1 + 37.6 $\log_{10}(d)$ dB
Shadowing	Log-normal (6dB)
Traffic characteristics	
Users' arrivals	Log-normal with mean λ and variance $v = \frac{\lambda}{10}$
Service type	FTP
Average file size	4 Mbits

We consider in this study a BS having a very low load. We consider a mean arrival rate $\lambda = 1$ user/s/km^2 which translates into a load of around 1%, i.e., 1% of the time during the simulation the BS is serving users and it is idle for the remaining time. It is in very low load that the ASMs can significantly impact EC and delay.

The power figures for the different states of the BS with 1 sector are given in Table III. These power values are deduced using IMEC power model tool [14].

TABLE III: Power consumption of 2x2 MIMO BS (1 sector) in different states.
Radiated Power: 46 dBm, Bandwidth: 20 MHz

Active	**Idle**	**SM$_1$**	**SM$_2$**	**SM$_3$**
250 W	109 W	52.3 W	14.3 W	9.51 W

The Q-learning parameters are taken as follows: $\gamma = 0.1$; $\epsilon_{exp} = 0.1$ and $\alpha = \frac{1}{N_{s,a}}$; where $N_{s,a}$ is the number of visits of state-action pair (s, a).

We define the sets of possible actions as:
N_3 = {100, 500, 700, 1000, 1500, 2000},
N_2 = {1000, 2000, 3000, 5000, 10000, 20000},
N_1 = {10000, 20000, 50000, 100000, 150000, 200000}.

The rational behind the choice of these actions sets is the following: the number of consecutive SMs of a given level should span the time interval covered by the SM of the lower level. These values are taken in such a way that we can get different policies according to the defined reward function and in order to have a finite set of possible actions.

B. Convergence analysis

At each decision point, we compute the maximum variation of the quality value $Q(s, a)$ for all the state-action pairs (s, a). This variation tends to zero for a sufficient period of learning and this defines our convergence criteria. As an example, we take the case of a reward function with $\epsilon = 0.8$. Figure 2 shows the convergence behaviour of the learning phase in this configuration. We can observe that for sufficiently high number of iterations (in the order of 10^4), the maximum variation is nearly zero for all state-action pairs.

Once the convergence is assured, we can exploit the results of the learning phase in order to quantify the outcome of the selected policy.

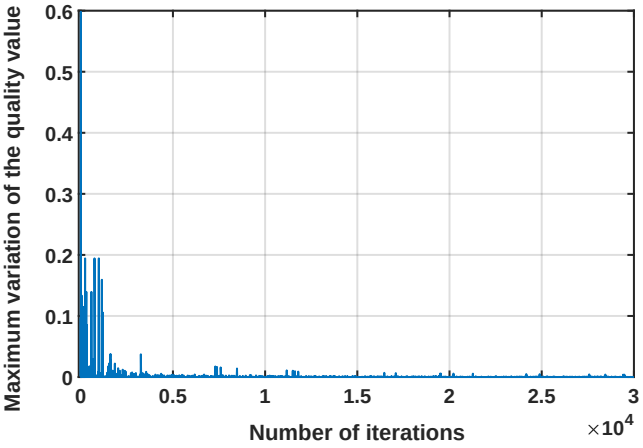

Fig. 2: Convergence evolution

C. Numerical results

The proposed approach is tested for different values of ϵ defining different tradeoffs between the EC reduction and the delay. This allows to learn different policies as presented in Table IV.

TABLE IV: Policies

	Policy (n_3, n_2, n_1)
$\epsilon \approx 0$	(2000, 20000, 200000)
$\epsilon = 0.2$	(1500, 20000, 200000)
$\epsilon = 0.5$	(1000, 20000, 200000)
$\epsilon = 0.8$	(1000, 3000, 200000)
$\epsilon \approx 1$	(700, 1000, 200000)

After the learning phase, we exploit the derived policy and assess its performance in terms of EC reduction and delay increase. Our baseline scenario is when we do not use any SM.

Figure 3 presents the repartition of the different states of the BS as a function of ϵ. We can see that for $\epsilon \approx 0$ (the priority is to maximize the energy gain), SM_3 is used almost all the time (98% of time). Whereas, when $\epsilon \approx 1$ (we have a high constraint on the delay), SM_3 is less used (50% of time) and more time is given to SM_2 then SM_1 before waking up to anticipate the arrival of a user. In between, as ϵ increases we shift gradually from SM_3 to the two other levels.

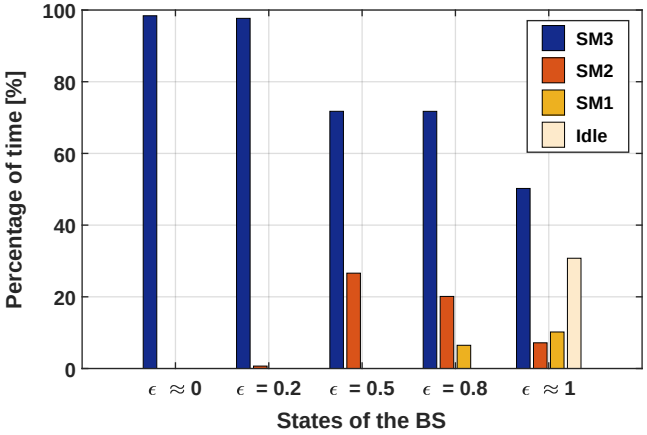

Fig. 3: Hystograms for the different SM policies as a function of ϵ

Figure 4 shows both the energy gain and the delay increase resulting from each selected policy.

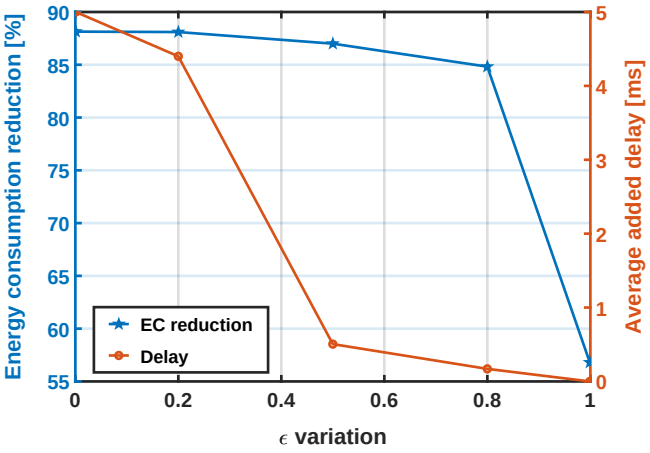

Fig. 4: Performance assessment of the selected policies during the exploitation phase

We can see that the energy gain and the delay depend on the chosen ϵ, i.e., the reward function. The lower ϵ, the higher the energy gain because deeper SMs are allowed to be repeated more times. This is translated also into a higher increase of the delay. Thus, the network operator has to choose carefully this parameter in order to satisfy the requirements of the different 5G use cases. For instance, in the case of URLLC, ϵ should be fixed to 1 which will prevent having an increase of latency but it is still energy-efficient since we can reduce up to almost 57% of the EC.

V. Conclusion

We proposed in this paper a management solution based on Q-learning approach enabling to orchestrate the ASM levels according to the requirements of the network operator in terms of EC reduction and delay. With this proposed approach, the control agent can decide how many times the BS can repeat each ASM level depending on the defined utility. Our results show that this solution is very efficient: when we have a high constraint on the delay, the agent learns the optimal policy enabling to have high energy gain without increasing the delay. In this case, we can reduce the EC by 57%. Whereas, if no constraint is imposed on the delay, the BS stays in the deepest SM until the arrival of a user which allows energy savings of almost 90%. For the cases in between, a tradeoff is found between the EC reduction and the delay increase based on the reward function.

As the BS has to send periodically signaling bursts, the ASM implementation would be impacted by this periodicity which can prevent from going into some levels of ASMs. It is interesting to extend this work to derive the optimal policies that can be used with the different signaling periodicities allowed in 5G networks.

References

[1] J. G. Andrews et al., "What Will 5G Be?," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1065-1082, June 2014.

[2] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, "The global footprint of mobile communications: The ecological and economic perspective," IEEE Communications Magazine, vol. 49, no. 8, pp. 5562, August 2011.

[3] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," IEEE Communications Magazine, vol. 49, no. 6, pp. 3037, June 2011.

[4] S. Buzzi, C. L. I, T. E. Klein, H. V. Poor, C. Yang and A. Zappone, "A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 4, pp. 697-709, April 2016.

[5] P. Frenger, P. Moberg, J. Malmodin, Y. Jading and I. Godor, "Reducing Energy Consumption in LTE with Cell DTX," 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), Yokohama, 2011, pp. 1-5.

[6] J. Wu, Y. Zhang, M. Zukerman and E. K. N. Yung, "Energy-Efficient Base-Stations Sleep-Mode Techniques in Green Cellular Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 803-826, Secondquarter 2015.

[7] B. Debaillie, C. Desset and F. Louagie, "A Flexible and Future-Proof Power Model for Cellular Base Stations," 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, 2015, pp. 1-7.

[8] F. E. Salem, A. Gati, Z. Altman and T. Chahed, "Advanced Sleep Modes and Their Impact on Flow-Level Performance of 5G Networks," 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1-7.

[9] NGMN Alliance, 5G White Paper, 2015, https://www.ngmn.org/5g-white-paper/5g-white-paper.html

[10] O. G. Aliu, A. Imran, M. A. Imran and B. Evans, "A Survey of Self Organisation in Future Cellular Networks," in IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 336-361, First Quarter 2013.

[11] 3GPP TSG-RAN WG1 Meeting #88 R1-1703092, "On Requirements and Design of SS Burst Set and SS Block Index Indication, Athens, Greece 13th - 17th February 2017.

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

[13] C. J. Watkins and D. Peter, "Q-learning," Machine Learning, vol. 8, no. 3-4, pp. 279-292, 1992.

[14] IMEC Power Model Tool, http://www2.imec.be/be en/research/wirelesscommunication/ power-model-html.html/.