Network Selection in 5G Networks based on Markov Games and Friend-or-Foe Reinforcement Learning

Alessandro Giuseppi*, Emanuele De Santis, Francesco Delli Priscoli, Seok Ho Won, Taesang Choi,
Antonio Pietrabissa

Abstract— **This paper presents a control solution for the optimal network selection problem in 5G heterogeneous networks. The control logic proposed is based on multi-agent Friend-or-Foe Q-Learning, allowing the design of a distributed control architecture that sees the various access points compete for the allocation of the connection requests. Numerical simulations validate conceptually the approach, developed in the scope of the EU-Korea project 5G-ALLSTAR.**

Keywords— Multi-Agent Reinforcement Learning, 5G, Network Selection, Markov Games

I. Introduction

The problem of Network Selection arises in the framework of the so-called “heterogeneous networks”, modern communication scenarios in which several different Radio Access Technologies (RATs) are available to connect a user with the Core Network (CN). In such networks, when a new connection is established, a decision regarding which Access Point (AP) to utilize shall be taken by either the UE or the network itself, based on a feedback-based analysis of the network state.

Different criteria (e.g., congestion state, power efficiency, reliability) may be utilized for the selection, and, based on the scope of the information gathered for the analysis, it is possible to identify three different classes of approaches[1]:

- *User-Centric Approach*: in which the User Equipment (UE) monitors the APs state and takes its connection decisions based on some thresholds-based performance parameters (e.g. Signal to Noise Ratio) measurable locally. In advanced scenarios, the UE could consider other RATs characteristics (e.g. coverage, user preferences, …) to better satisfy the application and user needs.
- *RAN-Assisted Approach*: in this approach, an information exchange is done between the AP and the UE, so that the latter can select the one it prefers based on a broader feedback that may capture aspects not locally measurable, such the congestion level on the specific RATs, their expected resource allocation and their predicted/historical connection reliability.
- *RAN-Controlled Approach*: the previous approaches were user-centric by nature, and consequently could only attain a sub-optimal solution to the network selection problem, in this approach the decision is taken directly by the Radio Access Network (RAN), a controller that oversees the functioning of the various RATs that constitute the Access Network. The decision taken by the RAN can either be centralised or distributed, as the RAN itself that may have some functionalities distributed over the various RATs. In this approach, the UEs may be configured to report radio measurements on their local radio environment to integrate the feedback available to the centralised network controller. This solution is the one adopted by 3GPP for addressing dual-connectivity issues.

The solution presented in this paper can be classified as a control strategy of the RAN-Controlled category, characterised by the distribution of the control logic over the controller of the various RATs controllers that regulate the APs connection admittance logics, so that the network resources available for the connection are optimally exploited.

From a methodological point of view, several approaches were investigated in the literature for the network selection problem, spacing from solutions based on Multiple Attribute Decision Making (MADM) [2], [3], to Fuzzy Logic control systems [4], and Game Theory-based approaches [5], [6]. Additionally, Markov Decision Processes (MDPs) and Reinforcement Learning (RL) were tested, among the others, in [7] and [8].

The proposed approach utilizes results from both RL and Game Theory in a multi-agent framework. The problem will be modelled in such a way that the distributed RAT controllers will compete among each other for being selected to serve the connection requests. The overall goal of the control strategy will

This work was supported by the European Commission in the framework of the H2020 EU-Korea project 5G-ALLSTAR (5G AgiLe and fLexible integration of SaTellite And cellular, www.5g-allstar.eu/) under Grant Agreement no. 815323.

Alessandro Giuseppi, Emanuele De Santis, Francesco Delli Priscoli and Antonio Pietrabissa are with the Department of Computer, Control, and Management Engineering“Antonio Ruberti”, University of Rome La Sapienza, via Ariosto 25, 00162, Rome, Italy, and with the Space Research Group of the Consortium for the Research in Automation and Telecommunication (CRAT), via G. Nicotera 29, 00195, Rome, Italy.

Seok Ho Won and Taesang Choi with the Future Mobile Communication Lab and the Intelligent Network Research Lab of ETRI, Daejoen, Republic of Korea.

be the optimal usage of network resources, without relying on centralized control approaches – as, for example, with a common least loaded allocation logic which assigns the upcoming connections to the RAT with the lowest resource usage. In this regard, the present paper employs the so-called "friend-or-foe Q-learning" algorithm to govern the network according to an adversarial Nash strategy.

II. Preliminaries on Learning Markov Games

A. Markov Games and Nash Equilibria

A Markov Game among N players is defined as the tuple [10] $\langle S, \mathcal{A}, T, \mathcal{R}, \gamma \rangle$, where:

- S is the finite state space.
- $\mathcal{A} = \{A_i,\ i = 1, \dots ,\ N\}$ is the collection of the action sets available to the various players i.
- $T(s, a_1, a_2, \dots, a_N, s')\colon\colon S \times A_1 \times A_2 \times \dots \times A_N \times S \to \mathbb{R}$ is the state transition function, which describes the transaction probability between the two states s and s' when the agents take the actions $a_1, a_2, \dots, a_N$.
- $\mathcal{R} = \{R_i(s, a_1, a_2, \dots, a_N)\colon S \times A_1 \times A_2 \times \dots \times A_N \to \mathbb{R},\ A_i = \{a_i\}\}$ is the collection of reward functions that attribute a reward to each agent when they take actions $a_1, a_2, \dots, a_N$ and the system is in state s.
- $0 \leq \gamma < 1$ is the discount factor that captures the trade-off between short-term and long-term performances sought by the agents.

In this work we consider the so-called *general sum* games, meaning that no assumption is made on the cumulative reward attained by the agents, contrary to zero-sum games.

A policy $\pi_i(s)\colon S \to \mathbb{R}^{\#(A_i)}$ is a function that maps the state of the system into a probability distribution over the actions of player i. Each player is associated with a (state,action)-value function Q_i [11], [12], defined as

$$Q_i(s, a_1, a_2, \dots, a_N) = R_i(s, a_1, a_2, \dots, a_N) + \\ + \gamma \textstyle\sum_{s'} T(s, a_1, a_2, \dots, a_N, s') Q_i(s', \pi_1, \pi_2, \dots, \pi_N), \qquad (1)$$

in which $Q_i(s', \pi_1, \pi_2, \dots, \pi_N)$ is the weighted sum of the Q_i's according to the policies π_i 's. For their definition, the (state,action)-value functions represent the expected discounted reward attained over time by the players starting from state s, taking actions $a_1, a_2, \dots, a_N$ and following the policies $\pi_1, \pi_2, \dots, \pi_N$ from there on. The goal of the controllers that will determine the policy of each agent is the one of maximizing its own value function *unilaterally* (i.e., without cooperation).

An important concept to introduce in the framework of Markov Games is the one of adversarial Nash equilibria, which are a set of policies $\pi_1, \pi_2, \dots, \pi_i', \dots, \pi_N$ characterized by the following two properties [12]:

- no player can improve its policy unilaterally, i.e.,

$$R_i(s, \pi_1, \pi_2, \dots, \pi_i, \dots, \pi_N) \geq R_i(s, \pi_1, \pi_2, \dots, \pi_i', \dots, \pi_N);$$

- no player sees its reward lowered by a change in the policies of the other players, i.e.,

$$R_i(s, \pi_1, \pi_2, \dots, \pi_i, \dots, \pi_N) \leq R_i(s, \pi_1', \pi_2', \dots, \pi_i, \dots, \pi_N').$$

B. Multi-Agent Reinforcement Learning

In scenarios in which the agents are not provided with a complete and accurate model of the system, model-free control solutions as Reinforcement Learning (RL) [11] have to be implemented to attain the desired system behaviour.

The attractiveness of RL in Multi-agent scenarios is due to the fact that it allows the agent i behaviour, described by its policy π_i, to adapt to the strategy employed by the other agents. This capability becomes of crucial importance when the various agents compete one against each other, as each agent has no incentive to share information regarding its own configuration with the others. Nevertheless, RL also allows the agent to learn about the environment characteristics by directly interacting with it, meaning that no explicit knowledge of the functions T and $R_j, j = 1, \dots, N$ is assumed or necessary for reaching an optimal control strategy.

In this paper, the Friend-or-Foe Q-Learning algorithm from [12] is employed in its adversarial variant. The additional degree of information that agent i requires other than the feedback observation of the tuple $\langle s, a_1, \dots, a_N, s', r_i \rangle$ is the classification of the other players as either friends (cooperating agents that try to maximise their rewards jointly) or foes (competing agents that try to maximise their own rewards unilaterally and, consequently, to minimise player i's reward).

In this algorithm, each agent i learns its (state,action)-value function Q_i according to the following rule:

$$Q_i(s, a_1, a_2, \dots, a_N) = \big(1 - \alpha(t)\big) Q_i(s, a_1, a_2, \dots, a_N) + \\ + \alpha(t)\big(r_i + \gamma\, Nash_i(s, Q_1, Q_2, \dots, Q_N)\big), \qquad (2)$$

where $Nash_i(s, Q_1, Q_2, \dots, Q_N)$ is computed as ([12])

$$Nash_i(s, Q_1, Q_2, \dots, Q_N) = \max_{\pi \in \Pi(A_1 \times \dots \times A_k)} \min_{[a_k, \dots, a_N] \in (A_{k+1} \times \dots \times A_N)} \\ \textstyle\sum_{[a_k, \dots, a_N] \in (A_{k+1} \times \dots \times A_N)} \pi(a_1) \cdots \pi(a_k) Q_i(s, a_1, \dots, a_N). \qquad (3)$$

In (3), it is assumed, without loss of generality, that the players $1, \dots, k$ cooperate with agent i and players $k + 1, \dots, N$ are its foes. The sequence $0 \leq \alpha(t) < 1$ represents the evolution, over time, of the *learning rate* of the agents. Under the hypothesis that that $\sum_t \alpha(t) = +\infty$ and $\sum_t \alpha(t)^2 < +\infty$ [11], Theorem 6 in [12] proves that Foe Q-Learning (i.e., in the case in which all agents are foes) converges to an adversarial equilibrium, provided that such an equilibrium exists.

III. Modelling Network Selection as a Markov Game

As already introduced, the network selection problem will be modelled as a Markov Game in which each AP is a competing player.

A. State Space

The state of the network can be represented by the percentage of occupied resources on each of the APs. In order to have a finite number of states, a possible solution is to quantize the percentage of resources with a factor q. The set of states is then defined as:

$$S = \left\{[s_1, s_2, \dots, s_n], s_i = nq,\ 0 \le s_i \le 1, n \in \{0, \dots, 1\} \cdot \frac{100}{q} \right\},$$

meaning that there are $(q+1)^n$ different states.

B. Action Space

The actions available to each of the agents regard the decision of whether to accept or decline the allocation of the incoming connection. Assuming that m different service classes are available to network users, a total of 2^m actions are required to model all the possible different choices. Note that some actions might be unavailable since the APs could decide to accept only services of certain classes.

The action set of user i is then defined as:

$$A_i = \left\{[a^1, a^2, \dots, a^m],\ \ a^j \in \{0,1\}\right\}.$$

C. Reward functions

The reward that is given to each agent i for successfully allocating a service of class j depends on the service characteristics and on the amount of resources involved in the allocation.

Assuming that the service requires t_j resources, it is possible to model the reward as

$$r_i = \alpha_{ij} t_j \frac{B_i + t_j}{C_i},$$

where B_i and C_i represent the amount of resources occupied on AP i before the new allocation and the total capacity of the AP i, respectively. The factor α_{ij} serves the purpose of prioritizing certain services over other ones and/or modelling the fact that some APs are more appropriate, in terms of Quality of Service, for certain services.

The structure of the reward allows incentivizing the agent to allocate all of their resources, while also dedicating them to the most prioritized services.

When a new service request arrives at the agents, each of them selects its action and consequently takes the decision of being available for the allocation or not. One agent is sampled randomly from the list of available ones, and the allocation procedure proceeds. The agents that offered their availability to allocate the incoming service but were not selected for the actual allocation receive a small negative reward to disincentivize the behaviour of always offering the allocation availability. Furthermore, an agent that offered the allocation but was not able to fulfil it due to a scarcity of resources is given a highly negative reward to penalize its behaviour and the connection is discarded.

To avoid that all agents reject the less rewarding services, a negative reward is also given to all the agents if no agent offers its availability for the new allocation.

D. ε-Greedy Policy Selection

A fundamental concept in RL is the trade-off between knowledge exploitation and environment exploration. The update of the Q_i tables (2) and the solution of the *maximin* problem (3) represent, respectively, the process of learning from experience, or knowledge acquiring, and its exploitation to derive a proper strategy for the player. To provide the players with an adequate degree of exploration, the action selection is subject to the following rule, known as ε-gready selection:

$$a_i = \begin{cases} \text{argmax}_{a_i}(Nash_i), \text{with probability } 1-\varepsilon \\ \text{randomly chosen in the set } A_i, \text{with probability } \varepsilon \end{cases}, \quad (4)$$

where $Nash_i$ is the operator described in (3), in the case in which all the Aps are assumed to compete one with each other and, hence, there is no friend player that cooperates with the agent i. As suggested in [13], a possible refinement to (4) is to consider a decreasing sequence of values for ε, modelling the fact that the agent benefits more from the exploration process at the beginning, while knowledge exploitation becomes more effective as the agent experienced the system evolution and its possible states several times.

E. Maximin Linear Programming Formulation

In general, a *maximin* optimization problem takes the following form [14]:

$$\begin{aligned} \max \min_{j=1,\dots,n} \ & J(x_j) = c_j x_j, \\ \text{s.t.} \quad & A_{eq}x = g, \\ & A_{ub}x \le b, \end{aligned}$$

where $c_j \ge 0$ are scalars, A_{eq}, A_{ub} are matrices and g and b vectors of appropriate dimensions.

It is well known that such a formulation is equivalent to the following LP problem [14]:

$$\begin{aligned} \max_{z \in \mathbb{R}} \ & J(x_j) = c_j x_j, \\ s.\text{t.} \quad & A_{eq}x = g, \\ & A_{ub}x \le b, \\ & z \le c_j x_j, j = 1, \dots, n, \end{aligned}$$

where z is a scalar unknown that is bounded by the smallest value $c_j x_j$ by means of the additional third constraint.

In the context of Foe-Q-Learning, the *maximin* problem that appears in (3) becomes

$$\max_{\pi \in \Pi(A_1 \times \dots \times A_k)} \ \min_{[a_k,\dots,a_N] \in (A_{k+1} \times \dots \times A_N)} \sum_{[a_1,\dots,a_k] \in A_1 \times \dots \times A_k} \pi(a_1) \cdots \pi(a_k)\, Q_i[s, a_1, \dots, a_N]$$

$$\begin{aligned} s.t. \quad & \pi(a_i) \ge 0 \ \forall\, a_i \in A_j, j = 1, .., k \\ & \textstyle\sum_{i=1}^{k} \pi(a_i) = 1, \end{aligned}$$

leading to an equivalent LP formulation of the form:

$$\max_{\pi\in\Pi(A_1\times\ldots\times A_k)} z$$

$$\text{s.t.}$$

$$h_i = \sum_{[a_1,\ldots,a_k]\in A_1\times\ldots\times A_k} \pi(a_1)\cdots\pi(a_k)\, Q_i[s,a_1,\ldots,a_N],$$

$$\forall\, a_k,\ldots,a_N \in (A_{k+1}\times\ldots\times A_N),$$

$$\pi(a_i) \geq 0\ \forall\, a_i \in A_1,..,A_k$$

$$\sum_{i=1}^{k} \pi(a_i) = 1$$

$$z \leq h_i\ \forall\, [a_k,\ldots,a_N] \in (A_{k+1}\times\ldots\times A_N).$$

The following table reports the pseudo-code of the Network Selection Algorithm.

Table 1 Algorithm Pseudo-Code

```
Initialize $Q_i(s,a), i = 1,\ldots,n$, arbitrarly.
For each connection request (episode) do:
    • Each player observes the state of the system and the class $j$ of the upcoming connection
    • Each player selects its action $a_i$ with an $\varepsilon$-Greedy policy based on their $Nash_i$ function
    • The connection is allocated using the resources of one of the players that selected an action with $a^j = 1$. If no player was available for the allocation one is selected randomly.
    • All players receive a reward $r_i$ as described in section III.C.
    • All players update their $Q_i$ table according to eq. (2).
end
```

IV. Simulations

A. Simulation setup

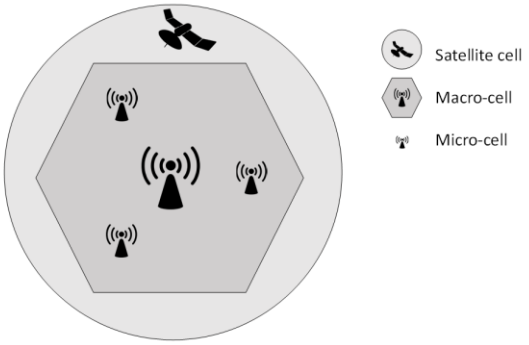

Figure 1 Connection Area covered by 3 different Radio Access Technologies

The scenario considered in the simulation is reported in Figure 1, and consists in an area covered by three different RATs. The number of agents considered is then $N = 3$. The resource considered for the connection is throughput, and each RAT had a maximum capacity of 1000 Mbps. Two service classes were modelled, the first characterised by a resource request of $t_1 = 1\,Mbps$ and the latter by $t_2 = 5\,Mbps$. The parbameters α_{ij} were set differently for each simulation.

A total of 2000 user requests were generated, where each request had a 0.8 probability of being a new connection and 0.2 of being the end of a connection, with a consequent resource deallocation. The connection requests were uniformly distributed over the two service classes.

Regarding the RL-based controller parameters, $\alpha(t)$ was set as $\alpha(t) = 1/(1 + \lfloor t/10 \rfloor)$, where $\lfloor\ \rfloor$ represents the lower-integer operator, and $\varepsilon(t)$ halved every 100 iterations starting from $\varepsilon(0) = 0.6$. Finally, the discount factor γ was set to 0.9.

The simulative scenario considered is a simplified one, but maintains the dimensioning and the key characteristics of the test cases that are envisaged to be developed in the scope of the 5G-ALLSTAR project.

B. Simulation one – baseline Least Loaded Controller

The baseline controller considered as a benchmark follows a least-loaded AP logic, as it assigns the upcoming connections to the APs with the lowest relative resource usage. Such a controller is centralised by nature, as it requires a complete knowledge of the state of the system.

C. Simulation two – no service prioritisation

In this simulation it was assumed $\alpha_{ij} = 1\ \forall i,j$, meaning that the reward of the agents depends only on the amount of allocated resources and no priority was given to any of the two service classes.

D. Simulation three – different rewards

The controllers trained in this simulation received a reward for the allocations characterised by $\alpha_{i,1} = 2$ and $\alpha_{i,2} = 0.2, \forall i$. This choice makes the per-bps reward higher for the first class of service.

E. Simulation results

From the analysis of Figure 2 and Figure 3, it is possible to note how the two RL agents behave differently from the centralised Least Loaded controller. In particular, the controller of simulation 2 tends to uniformly accept the two services, in line with the fact that they were characterised by the same amount of reward per-Mbps, while the second RL controller (simulation three) favours the allocation of services of the first class. Overall, both the Least Loaded controller and the controller of simulation 2 blocked a total of 409 Mbps, meaning that the first RL solution fully exploits its available resources. The second RL controller, on the contrary, allocates a slightly lower amount throughput, blocking a total of 463 Mbps. This different behaviour is due to the fact that the agents obtain, for the same amount of resources, a different pay-off depending on the service class. In fact, due to the choice of the parameters α_{ij}, the services of the first class provide ten times the amount of reward per Mbps with respect to the other. Even if the second RL controller blocked more Mbps, this translated in an improvement in performances, measured in terms of its cumulative total reward, of approximately 10%.

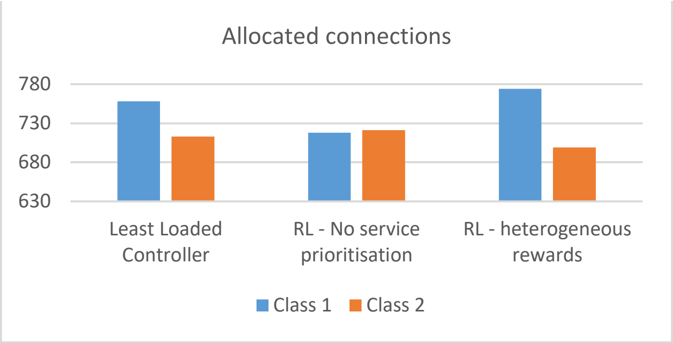

Figure 2 Number of allocated connections for the three controllers, divided by service class

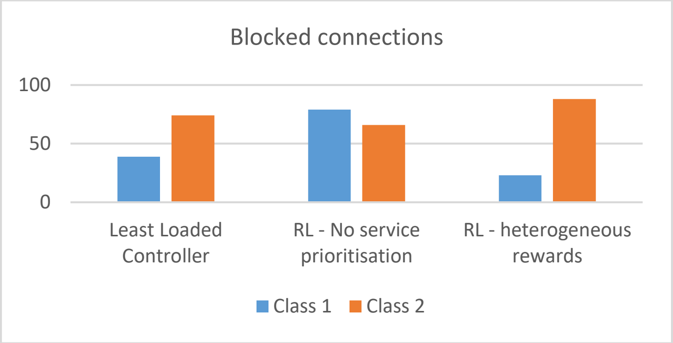

Figure 3 Number of blocked connection for the three controllers, divided by service class

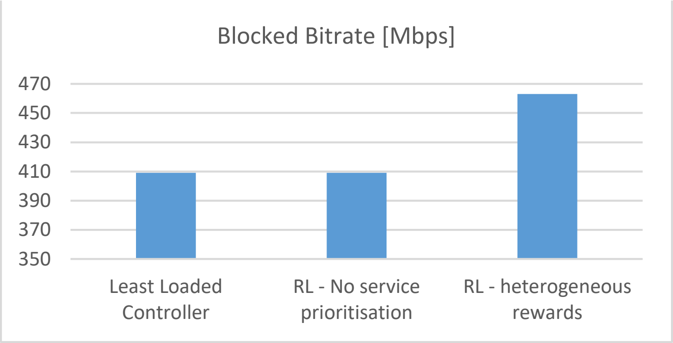

Figure 4 Total amount of bitrate of the blocked connections

V. Conclusions and Future Works

The paper presented a distributed control approach for the problem of network selection. The proposed solution was based on Friend-or-Foe Q-learning, a multi-agent distributed Reinforcement Learning approach to solve Markov Games. The problem was modeled as a standard multi-agent Markov Decision Problem, and an adversarial game was formulated. The preliminary simulations presented validated the concept of the approach, while future testing on more realistic scenarios will be carried out within the scope of the H2020 5G-ALLSTAR project.

VI. Acknowledgements

The authors acknowledge the researchers of the CRAT and ETRI teams involved in the 5G-ALLSTAR project and all the consortium members involved in Work Package 4.

References

[1] S. Andreev *et al.*, “Intelligent access network selection in converged multi-radio heterogeneous networks,” *IEEE Wirel. Commun.*, vol. 21, no. 6, pp. 86–96, Dec. 2014.

[2] L. Hui, W. Ma, and S. Zhai, “A novel approach for radio resource management in multi-dimensional heterogeneous 5G networks,” *J. Commun. Inf. Networks*, vol. 1, no. 2, pp. 77–83, Aug. 2016.

[3] N. Zhang, S. Zhang, S. Wu, J. Ren, J. W. Mark, and X. Shen, “Beyond Coexistence: Traffic Steering in LTE Networks with Unlicensed Bands,” *IEEE Wirel. Commun.*, vol. 23, no. 6, pp. 40–46, Dec. 2016.

[4] A. Wilson, A. Lenaghan, and R. Malyan, “Optimising Wireless Access Network Selection to Maintain QoS in Heterogeneous Wireless Environments,” in *International Symposium on Wireless Personal Multimedia Communications 2005 (WPMC 2005)*, 2005, pp. 1236–1240.

[5] M. Cesana, N. Gatti, and I. Malanchini, “Game Theoretic Analysis of Wireless Access Network Selection: Models, Inefficiency Bounds, and Algorithms,” in *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, 2008, p. 6.

[6] J. Antoniou and A. Pitsillides, “4G Converged Environment: Modeling Network Selection as a Game,” in *2007 16th IST Mobile and Wireless Communications Summit*, 2007, pp. 1–5.

[7] X. Gelabert, J. Perez-Romero, O. Sallent, and R. Agusti, “A Markovian Approach to Radio Access Technology Selection in Heterogeneous Multiaccess/Multiservice Wireless Networks,” *IEEE Trans. Mob. Comput.*, vol. 7, no. 10, pp. 1257–1270, Oct. 2008.

[8] N. Vučević, J. Pérez-Romero, O. Sallent, and R. Agustí, “Reinforcement learning for joint radio resource management in LTE-UMTS scenarios,” *Comput. Networks*, vol. 55, no. 7, pp. 1487–1497, May 2011.

[9] 5GPPP Architecture Working Group, “5GPPP Architecture Working Group View on 5G Architecture,” 2017.

[10] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Machine Learning Proceedings 1994*, 1994, pp. 157–163.

[11] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[12] M. L. Littman, “Friend-or-Foe Q-learning in General-Sum Games,” 2003.

[13] F. Liberati *et al.*, “Stochastic and exact methods for service mapping in virtualized network infrastructures,” *Int. J. Netw. Manag.*, vol. 27, no. 6, p. e1985, Nov. 2017.

[14] R. K. Ahuja, “Minimax linear programming problem,” *Oper. Res. Lett.*, vol. 4, no. 3, pp. 131–134, 1985.