NEGOTIATION STRATEGIES TO IMPROVE DISTRIBUTED POWER ALLOCATION FOR SELF-ORGANIZING HETEROGENEOUS NETWORKS

Juan Parras and Santiago Zazo

Information Processing and Telecommunications Center
Universidad Politécnica de Madrid
Madrid, Spain

ABSTRACT

As the traffic supported by wireless networks grows, new solutions to increase the throughput are emerging, such as heterogeneous networks, which face the challenge of self-organization in order to control the interference. A frequently used tool to model this problem is Game Theory, and we show that Communicate & Agree, a recent general-purpose negotiation-based algorithm for solving repeated games, can be applied to these network settings to provide Pareto-efficient results. As a particular application, we study a distributed power allocation problem, showing that Communicate & Agree provides better payoffs than a state-of-the-art baseline while being faster, thus making it an ideal candidate for self-organizing heterogeneous networks.

Index Terms— Repeated games, Self-organizing networks, Heterogeneous networks, Communicate & Agree, Distributed power allocation

1. INTRODUCTION

In the past years, the traffic supported by wireless networks has grown exponentially due to the massive increase in both the number of devices interconnected and the amount of data transmitted. Moreover, both are expected to continue to grow in the incoming years: more and more devices are connected under the paradigm of the Internet of Things, and each device transmits more and more data, due to the new services that require a significant transmission rate. The traffic rise puts the network infrastructure under pressure, and hence, new concepts are emerging to address this challenge, such as Heterogeneous Networks.

An Heterogeneous Network may combine different cell types, from macrocells to nano, pico and femtocells, in order to give different kinds of service to different users thanks to the possibilities of spatial spectrum reuse. However, the flexibility provided by the different cell sizes comes at the cost of an increase in the difficulty to organize, manage and optimize the network resources. In order to automatize these procedures, there is a lot of ongoing research in the direction of self-organizing networks, which aims to automatically adjust the network with minimal, if any, human intervention. Artificial intelligence techniques are being used in this area [1], and among these, a set of techniques that stand out as a promising framework are based on Game Theory [2], [3], [4], [5], which is the branch of mathematics that models conflicts between players.

Among all the different kind of games that exist [6], Repeated Games (RGs) are frequently used in network environments. In an RG, an interaction between several players is repeated a certain number of times, where each interaction is called stage. The solution concept is called equilibrium: an action that provides a reward that no player can unilaterally improve. RGs are characterized by having a reward function that does not change during the stages, and hence, RGs adapt adequately to network problems as there are many interactions between network nodes that happen more than once without any change in the interest of each node. For instance, RG tools have been applied to resource allocation [3], [5], interference management [7] or cyberdefense [8]. However, it is frequent that each work proposes its own method to obtain a valid equilibrium, which means that most of these algorithms cannot be used even in similar settings due to the fact that they exploit the assumptions and structures of the concrete problem for which they were designed [3], [7], [9].

In this work, we propose using a recent, general-purpose algorithm to obtain RG equilibria, called Communicate & Agree (CA) [10]. It is based on negotiation, requiring that different players communicate, thus it can be implemented in a straightforward way in a network setting. Being a general-purpose algorithm, it can be implemented for any RG with discount. We show the capacities of CA in this work by applying it to a distributed power allocation problem: our simulations show that CA not only provides better results than a specific state-of-the-art algorithm designed for this concrete problem, but it is also faster, and hence, it constitutes a very promising tool for self-organizing networks.

The rest of the paper goes as follows: in Section 2 we introduce the Game Theory background needed, including a description of CA. Then, Section 3 introduces the distributed power allocation setting that we use to test our ideas. Later, Section 4 contains validation simulations. Finally, Section 5 draws some conclusions and proposes several possible future lines.

2. GAME THEORY BACKGROUND

A static game is defined as follows [11]:

Definition 1. *A static game G is a triple $\langle N_p, A, r\rangle$, where:*

- *N_p is the number of players.*
- *A_i is the set of actions available to player i, with $A = \prod_i A_i$ being the set of actions available to all players.*
- *$r : A \to \mathbb{R}^{N_p}$ is a function that gives the game rewards.*

An RG is defined by using a static game as follows [12]:

Definition 2. *A Repeated Game (RG) is built by using a static game (also called stage game) which is played repeatedly infinite times. Its main components are the following, where superscript indicates time and subscript indicates the players:*

The work is supported by the Spanish Ministry of Science and Innovation under the grant PID2020-112502RB-C41 (NAUTILUS).

1. *The set of histories $\mathscr{H}^t \equiv A^t$. A history h^t is a list of actions played in periods $[0, .., t-1]$.*
2. *A strategy for player i is a mapping from the set of all possible histories into the set of actions: $\sigma_i : \mathscr{H} \rightarrow A_i$. σ denotes the strategy of all players.*
3. *The discounted payoff to player i is obtained as:*

$$V_i(\sigma) = (1-\delta) \sum_{t=0}^{\infty} \delta^t r_i^t(a^t(\sigma)), \tag{1}$$

where $\delta \in (0,1)$ is the discount factor, $a^t(\sigma)$ denotes that action $a^t = (a_i, a_{-i})$ is chosen following strategy $\sigma = (\sigma_i, \sigma_{-i})$ and the subscript $-i$ refers to all players except player i.

The discount factor can be understood as a measure of the patience of the players [12] or as a measure of the uncertainty in the temporal horizon, i.e., the RG may end at any stage with probability $1-\delta$ [13], so the game has an expected number of stages equal to $(1-\delta)^{-1}$. In a network setting, the latter is more useful, as it allows us to model the fact that the network interaction may end, while the precise duration of the interaction is unknown a priori.

The best known solution concept to a game is the Nash Equilibrium (NE), which mathematically is expressed as:

$$V_i(\sigma_{i,NE}, \sigma_{-i}) \geq V_i(\sigma_i, \sigma_{-i}), \quad \forall \sigma_i, \quad \forall i \in \{1, ..., N_p\},$$

and it means that no player can get a better payoff by a unilateral deviation. Every static game has at least one NE [14], and every NE of the stage game is a NE of the RG, so every RG has at least one NE. However, as $\delta \rightarrow 1$, there might appear other NEs due to the Folk Theorem [12], [6], and these new NEs may provide better payoffs to all players compared to the stage game NE. Thus, the use of strategies that make use of the Folk Theorem are very convenient for efficiency, as they may improve the payoffs for all players.

A strategy that is frequently used to take advantage of the Folk Theorem is Grim trigger. Under this strategy, there are two actions a_o and a_p, where a_o provides a good payoff to all players, and a_p is a punishment strategy such as an NE of the stage game. When all players are following a Grim trigger strategy, they compromise to play a_o, and in case that any player deviates, all players switch forever to a_p. Grim trigger is a suitable strategy if the gains of deviating from a_o and receiving a_p do not surpass the gains of not deviating. Mathematically, that means [10]:

$$\begin{aligned} &(1-\delta) r_i(a_o) + \delta V_i(a_o) \geq \\ &(1-\delta) \max_{a_i' \neq a_{i,o}} r_i(a_i', a_{-i,o}) + \delta V_i(a_p), \quad \forall i \in \{1, ..., N_p\}. \end{aligned} \tag{2}$$

If a player uses action a forever, then we have that $V_i(a) = r_i(a)$, which allows reformulating (2) as:

$$r_i(a_o) \geq (1-\delta) \max_{a_i' \neq a_{i,o}} r_i(a_i', a_{-i,o}) + \delta r_i(a_p), \quad \forall i. \tag{3}$$

2.1. CA algorithm

CA is a recent algorithm for obtaining an RG equilibrium [10], which is based on prior negotiation between the players to find a valid Grim trigger strategy. It has several features that make it ideal as a general algorithm for the kind of RGs that appear in networking settings: it is fully distributed, it is designed to work with discounted payoffs, it uses the Grim trigger strategy and it makes use of the Folk Theorem and selects payoffs that are Pareto efficient.

CA negotiation is based on two main steps. In the first step, denoted action space sampling, each player i randomly samples the action space A in order to obtain actions such that condition (3) is fulfilled for them. When a player finds such action, she shares it to the rest of the players, who also check whether that action fulfills (3) for them. When an action fulfills (3) for every player, it is stored as a candidate action for a_o. Note that in this step, all players only need to know their own reward function and a punishment strategy that may be an NE of the stage game. It is possible that the players are unable to find a candidate a_o: this may be due to the fact that a_p already provides the best payoff to all players, or that the players have not found an adequate candidate a_o during sampling. The latter can be solved by allowing the players to sample for longer: it means investing more resources on sampling and exchanging more messages between players, and thus, there is a tradeoff between the probability of finding candidates a_o and the computational load.

In a second step, called Pareto pruning, the final a_o is chosen in a distributed way. Among all candidates a_o obtained, the players select one which is Pareto efficient (i.e., an action such that there is no other that provides better payoffs to all players), and this a_o is used for the Grim trigger strategy.

Hence, note that CA allows negotiating a good payoff before starting the game. It is an ideal algorithm for a networking environment, as it only needs communication between the players, which are nodes of the network, as well as a stage game equilibrium, that can be either known a priori or learned [10], [15].

3. DISTRIBUTED POWER ALLOCATION SETUP

As we have seen in the previous Section, CA is an ideal algorithm for solving RGs in networks in general: we will now show that it is also ideal for self-organization of heterogeneous networks. Let us focus on a concrete setting, which is distributed power allocation in wireless networks. When we have some Base Stations (BSs) which serve several Users (Us), if BSs and Us are close enough, they will interfere with the others, causing a decrease in the Signal to Interference and Noise Ratio (SINR). Each node of the network should adjust its power in order to minimize the interference while being able to transmit. Note that this is a conflict that has been already modelled by using Game theory tools [16], [17], [18], [3] or [5].

Let us focus on a similar model to the one presented in [3] and [5], where we consider that there are K BSs, where each BS is indexed by $k \in \{1, 2, ..., K\}$. We assume that each BS serves a single U, so BS_k serves U_k. The transmission power of BS_k is denoted as p_k^{BS}, while the transmission power of U k is denoted as p_k^U: note that p_k^{BS} is the downlink power, while p_k^U is the uplink power. Thus, we have that the SINR at U_k can be computed as follows:

$$SINR_{U_k} = \frac{p_k^{BS} l_{BS_k, U_k}}{N_0 + I_{U_k}}, \tag{4}$$

where $l_{a,b}$ denotes the signal attenuation between the network nodes a and b (which may be BSs or Us), and N_0 is the thermal noise level in the receiver (we assume the same N_0 for all BSs and Us), and I_{U_k} denotes the interference in U_k which has the following expression:

$$I_{U_k} = \sum_{n \in \{1, \dots, K\}, n \neq k} p_n^{BS} l_{BS_n, U_k} + \gamma \sum_{n=1}^{K} p_n^U l_{U_n, U_k}, \tag{5}$$

where we assume that uplink and downlink may transmit at the same time in different channels, and we model the co-channel interference with a parameter $\gamma \in [0,1]$. Note that there are two terms in I_{Uk}:

the first accounts for the interference caused by all the BSs except for BS_k, and the second accounts for the co-channel interference caused by the uplink transmission powers (which includes the self interference of U_k to itself).

In an analogous way, the SINR at BS_k is:

$$SINR_{BS_k} = \frac{p_k^U l_{BS_k,U_k}}{N_0 + I_{BS_k}}, \quad (6)$$

where we have:

$$I_{BS_k} = \sum_{n \in \{1,\ldots,K\}, n \neq k} p_n^U l_{BS_k,U_n} + \gamma \sum_{n=1}^{K} p_n^{BS} l_{BS_n,BS_k}. \quad (7)$$

In order to study this setup, we may use Game Theory tools. Let us consider a game with $N_p = K$ players, where player k corresponds to BS_k. Let us assume that p_U^k are fixed and known $\forall k \in K$, and hence, the actions of each player correspond to p_k^{BS}, i.e., each BS adjust its transmission power. For simplicity, we assume that all players have a set of predefined transmission power levels which are equal for all players, i.e., $A_1 = A_2 = \ldots = A_K$. Finally, we can define the reward function for BS_k as follows:

$$r_k = \log(SINR_{U_k}) + \log(SINR_{BS_k}), \quad (8)$$

where we note that the reward of BS_k is the quality of their communication channel with U_k both in the uplink and downlink, and due to the Shannon–Hartley theorem, the reward is also related to the capacity of the communication channel.

It is important noting that the reward of BS_k is deeply interconnected to the actions of the other BSs. In order to see that, let us compute the partial derivatives of the reward r_k with respect to the actions, which are the following:

$$\begin{aligned} \frac{\partial r_k}{\partial p_k^{BS}} &= \frac{1}{p_k^{BS}} - \frac{\gamma}{N_0 + I_{BS_k}(p_k^{BS}, p_{-k}^{BS})} \\ \frac{\partial r_k}{\partial p_j^{BS}} &= -\frac{l_{BS_j,U_k}}{N_0 + I_{U_k}(p_{-k}^{BS})} - \frac{\gamma l_{BS_j,BS_k}}{N_0 + I_{BS_k}(p_k^{BS}, p_{-k}^{BS})} \\ &\forall j \in \{1, \ldots, K\}, j \neq k, \end{aligned} \quad (9)$$

where we have made explicit the dependency of I_{BS_k} with the action of all players, and I_{U_k} with the actions of the rest of players (remember that $-k$ indexes all players but k). By observing the expressions in (9), we can note that the partial derivative with respect to p_j^{BS} is always negative, i.e, if other BSs increase their transmission power, the reward for BS_k decreases. But at the same time, we can see that the partial derivative with respect to p_k^{BS} is positive if the following condition holds:

$$\gamma p_k^{BS} < N_0 + I_{BS_k}(p_k^{BS}, p_{-k}^{BS}), \quad (10)$$

which can be simplified by replacing (7) in order to obtain:

$$N_0 + \sum_{n \in \{1,\ldots,K\}, n \neq k} \left(p_n^U l_{BS_k,U_n} + \gamma p_n^{BS} l_{BS_n,BS_k} \right) > 0, \quad (11)$$

where (11) means that BS_k may increase its reward by increasing its transmission power as long as the sum of the thermal noise and the interference received from the rest of BSs is positive, which is a condition that is always fulfilled.

In other words, the reward of BS_k always increases by increasing its own transmission power p_k^{BS}, and decreases if any other BS increases its transmission power. This means that the described game is very competitive, and indeed, the only NE of the stage game consists on each BS transmitting with its maximum power [3], which also means maximizing the interference to the rest of players.

However, by considering an RG, which means that there is more than a single communication between each BS and its associated U, players may cooperate to obtain a better equilibrium thanks to the Folk Theorem. If all players follow a Grim trigger strategy, they may transmit with a power level that lowers the interference and increases the reward for all players if such transmission power exists. In order to enforce that strategy, if any BS is caught deviating (i.e., transmitting with a higher power), the rest of BSs switch to their maximum transmission power forever, where the deviation would be detected by checking whether the interference has increased.

This idea is followed in [3], where the transmission power a_o for the Grim trigger strategy is obtained by an initial phase of trial and error, where each BS transmits different power levels until they all are satisfied with the rewards they obtain. However, there are two main caveats to this algorithm: the first one is that it has a potentially high energy consumption during the training phase, as it may last a long time until all players converge to a good power level. The second problem is the duration of the training phase, which means that the communication system is not completely available while the players are finding a good transmission level. Also, in [5], an online method is used to play an RG without prior negotiation, trying to use the Folk Theorem to improve an NE. The main advantage of this method is the lack of training phase, but its main disadvantage is that it may yield results far from being Pareto-efficient.

We propose using CA in order to alleviate these problems: as CA is based on negotiation, the learning phase using CA consists on interchanging messages between BSs, which does not make the transmission system unavailable. Also, CA provides a Pareto-efficient payoff by design. And finally, the number of messages interchanged between BSs for negotiation can be adjusted, so the traffic generated can be controlled. Note also that this is an advantage in self-organizing networks, as when a new BS appears or disappears from the network, BSs only need to run a new negotiation in order to find an adequate transmission power level. And due to the fact that the discount factor can be used to account for the expected duration of the game, i.e., the expected time until a new player is added or removed, then the proposed scheme has clear advantages to be implemented in self-organizing networks.

4. EMPIRICAL VALIDATION

We validate our approach using simulations. We consider that we have $K = 2$ BSs, and we locate each node in the following plane coordinates: $(10, 10)$ for BS_1, $(0, 0)$ for BS_2, $(1, 8)$ for U_1 and $(5, 5)$ for U_2, where all positions are in meters. Then, we set $p_k^U = 10$ W, $k \in \{1, 2\}$, and consider that the transmission powers that each BS may choose are $p_k^{BS} = \{5, 10, 15, 20, 25, 30\}$ W, $k \in \{1, 2\}$. Regarding the rest of transmission parameters, we consider that the noise floor is $N_0 = 0.001$ W, the co-channel interference factor is $\gamma = 0.001$ and the path loss is computed as $a_{a,b} = d_{a,b}^{-4}$, where $d_{a,b}$ is the distance between a and b. Using these parameters, we can observe in Figure 1 that the NE is not Pareto efficient, as there are other transmission powers that provide a better payoff for both BS_1 and BS_2 (note that we choose $K = 2$ so that we are able to plot the payoff region).

In order to improve the payoffs, we compare CA with the algorithm proposed in [3], which we use as a baseline. In order to compare them, we first run the learning phase in the baseline and the negotiation phase in CA. For a fair comparison, we use a parameter

N_c, that represents the number of iterations of the learning phase in case of the baseline (i.e., the number of times that each player tests a transmission power level), and the number of communications between players in CA (i.e., the number of times that each BS proposes a set of transmission powers to the other BS). Then, we use the strategy proposed by each algorithm in the game and obtain the total payoff (1). The results obtained are in Figures 1 and 2.

First, in Figure 1, we show the payoffs that are available for both players for $\delta = 0.95$. We can clearly see that there are payoffs that are better than the NE to both players, and actually, these payoffs can be achieved by making use of a Grim trigger strategy. Thus, it is desirable that the network operates using these transmission powers, as it implies having lower interferences, and thus, increasing the network capacity. Figure 1 also shows how CA works: it first obtains a set of payoffs that are better than the NE for all players (action space sampling phase), and then, the players choose a payoff such that there is no other payoff that is better for both players. We emphasize that both steps are done in a completely distributed fashion, which is one of the main features of CA.

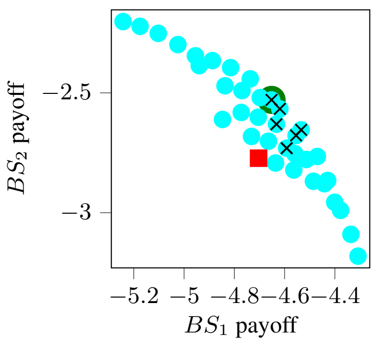
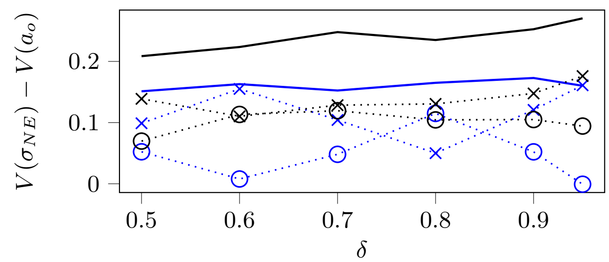
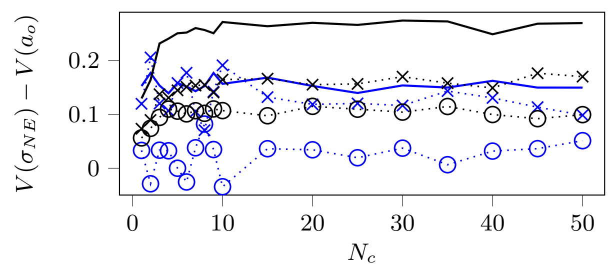


Fig. 1. Payoff region of the proposed game. In blue, we plot the possible payoff points given by all the combinations of transmission powers. The red square represents the NE payoff, obtained by transmitting at maximum power: note that there are payoffs that are better for both players. We also show an example of the results of each step of CA algorithm: the black crosses represent the candidate a_o that are sampled during the action space sampling step, where all of these actions provide a higher payoff than the NE. Among the candidate a_o set, CA selects an action that provides a payoff that cannot be improved for both players, which is the green point. For this plot, $\delta = 0.95$ and $N_c = 30$ for CA.

Second, in order to obtain the payoff for both players, we run CA and the baseline for a certain N_c value, and then use the strategy proposed to run a repeated game, which we truncate after 500 stages in order to compute the payoff (1). This procedure is repeated 50 times in order to average the results for each combination of δ and N_c. The results are in Figure 2, where the first subplot shows the comparison for different values of δ and the second subplot shows the comparison for different values of N_c. We remark that, in all cases, the payoff gain provided by CA with respect to the NE is positive (i.e., it always provides a better payoff than the NE for both players), and also, in all cases, CA provides a larger combined gain than the baseline for both players. Moreover, note that CA needs very few communications in order to achieve a good payoff (Figure 2, (b)), which means that CA negotiates fast and it introduces a very small overhead in the network traffic. CA is also faster than online learning algorithms, which need a training phase to obtain a good strategy (in a similar way to our baseline), and hence, they may take a long time to converge, such as [19], [20] or [21]. At the same time, CA provides better payoffs in the resource allocation setup than an online algorithm that does not need a training phase [5]. Thus, our empirical results suggest that CA is an ideal candidate for negotiating RG strategies in a networking setting, as the network nodes already communicate among them, and this facilitates the exchange of information for the negotiation phase.



(a) Payoff gain for δ values.



(b) Payoff gain for N_c values.

Fig. 2. Results comparison between CA and the baseline. We plot the payoff gain that each of these algorithms provides compared to the NE strategy. Baseline results are in blue, CA results are in black. Dashed lines represent the payoff of each BS: the circles are for BS_1 and the crosses, for BS_2, while solid lines are for the sum of the gains of both players. Note that CA always provides positive gains, which means that is always better than the NE, and also note that CA always provides a larger gain for both players than the baseline.

5. CONCLUSIONS

In this paper, we test CA as an algorithm for negotiating RG equilibria in network settings. We show that it is an ideal algorithm for self-organizing networks, as it provides good results in terms of payoffs, the traffic overhead that introduces can be adjusted, and it is fast to obtain valid strategies. We test CA on a distributed power allocation problem, and CA provides a significant gain in terms of payoffs to all players, which in our setting means decreasing the interference to other nodes and increasing the network capacity.

There are several topics that can be further explored. First, CA requires each player to know its own payoff function, but it may happen that they do not know it exactly, and thus, they would have an imperfect information game, which may change the negotiation process. And second, we have assumed that the network was static, i.e., the network nodes did not move, and there were no changes in the number of network nodes. As we have mentioned, it is possible to use δ in order to account for the uncertainties derived of these phenomena, and this is another interesting line of future research.

6. REFERENCES

[1] Xiaofei Wang, Xiuhua Li, and Victor CM Leung, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges," *IEEE Access*, vol. 3, pp. 1379–1391, 2015.

[2] Prabodini Semasinghe, Ekram Hossain, and Kun Zhu, "An evolutionary game for distributed resource allocation in self-organizing small cells," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 274–287, 2014.

[3] Prabodini Semasinghe and Ekram Hossain, "A cheat-proof power control policy for self-organizing full-duplex small cells," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[4] Tuğçe Bilen and Berk Canberk, "Overcoming 5g ultra-density with game theory: Alpha-beta pruning aided conflict detection," *Pervasive and Mobile Computing*, vol. 63, pp. 101133, 2020.

[5] Juan Parras, Patricia A Apellániz, and Santiago Zazo, "An online learning algorithm to play discounted repeated games in wireless networks," *Engineering Applications of Artificial Intelligence*, vol. 107, pp. 104520, 2022.

[6] Drew Fudenberg and Jean Tirole, *Game theory*, MIT press Cambridge, MA, 1991.

[7] Ehsan Monsef and Jafar Saniie, "A repeated game approach for interference mitigation of priority-based body area networks," in *2015 IEEE international conference on electro/information technology (EIT)*. IEEE, 2015, pp. 324–329.

[8] Juan Parras and Santiago Zazo, "Repeated game analysis of a csma/ca network under a backoff attack," *Sensors*, vol. 19, no. 24, pp. 5393, 2019.

[9] Zhengyang Qu, Di Chen, Gaofei Sun, Xinbing Wang, Xiaohua Tian, and Jing Liu, "Efficient wireless sensor networks scheduling scheme: Game theoretic analysis and algorithm," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 356–360.

[10] Juan Parras and Santiago Zazo, "A distributed algorithm to obtain repeated games equilibria with discounting," *Applied Mathematics and Computation*, vol. 367, pp. 124785, 2020.

[11] Tamer Basar and Geert Jan Olsder, *Dynamic noncooperative game theory*, vol. 23, SIAM, 1999.

[12] George J Mailath and Larry Samuelson, *Repeated games and reputations: long-run relationships*, Oxford university press, 2006.

[13] Dinh Thai Hoang, Xiao Lu, Dusit Niyato, Ping Wang, Dong In Kim, and Zhu Han, "Applications of repeated games in wireless networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2102–2135, 2015.

[14] John Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[15] Sergiu Hart and Andreu Mas-Colell, *Simple adaptive strategies: from regret-matching to uncoupled dynamics*, vol. 4, World Scientific, 2013.

[16] Wenmin Ma, Haijun Zhang, Wei Zheng, and Xiangming Wen, "Differentiated-pricing based power allocation in dense femtocell networks," in *The 15th International Symposium on Wireless Personal Multimedia Communications*. IEEE, 2012, pp. 599–603.

[17] Adnan Shahid, Saleem Aslam, Hyung Seok Kim, and Kyung-Geun Lee, "Distributed joint resource and power allocation in self-organized femtocell networks: A potential game approach," *Journal of network and computer applications*, vol. 46, pp. 280–292, 2014.

[18] Mohammed Al-Imari, Mir Ghoraishi, Pei Xiao, and Rahim Tafazolli, "Game theory based radio resource allocation for full-duplex systems," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015, pp. 1–5.

[19] Michael L Littman and Peter Stone, "A polynomial-time nash equilibrium algorithm for repeated games," *Decision Support Systems*, vol. 39, no. 1, pp. 55–66, 2005.

[20] Enrique Munoz De Cote and Michael L Littman, "A polynomial-time nash equilibrium algorithm for repeated stochastic games," in *Proceedings of the 24th Conference Annual Conference on Uncertainty in Artificial Intelligence*, 2008, pp. 419–426.

[21] Jacob W Crandall and Michael A Goodrich, "Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning," *Machine Learning*, vol. 82, no. 3, pp. 281–314, 2011.