A Multi-agent OpenAI Gym Environment for Telecom Providers Cooperation

Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, Benoit Radier

A Multi-agent OpenAI Gym Environment for Telecom Providers Cooperation

Tangui Le Gléau
Orange Labs
Lannion, France
tangui.legleau@orange.com

Xavier Marjou
Orange Labs
Lannion, France
xavier.marjou@orange.com

Tayeb Lemlouma
IRISA
Lannion, France
tayeb.lemlouma@irisa.fr

Benoit Radier
Orange Labs
Lannion, France
benoit.radier@orange.com

***Abstract*—The ever-increasing use of the Internet (streaming, Internet of things, etc.) constantly demands more connectivity, which incentivises telecommunications providers to collaborate by sharing resources to collectively increase the quality of service without deploying more infrastructure. However, to the best of our knowledge, there is no tool for testing and evaluating participation strategies in such collaborations. This article presents a new adaptable framework, based on the OpenAI Gym toolkit, allowing to generate customisable environments for cooperating on radio resources. This framework facilitates the development and comparison of agents (such as reinforcement learning agents) in a generic way. The main goal of the paper is to detail the available functionalities of our framework. We then focus on game theory aspects as multi-player games induced by these environments can be considered as sequential social dilemmas. We show in particular that although each agent has no incentive to remain cooperative at each step of such iterated games, a mutual cooperation provides better outcomes (in other words, Nash Equilibrium is non optimal)**

***Index Terms*—Multi-agent system, simulation frameworks, RAN sharing, Game Theory**

I. Introduction

More and more connectivity is necessary to respond to the needs due to the explosion of Internet uses (streaming, Internet of Things, autonomous vehicles, telemedicine etc). One major challenge of connectivity providers is to cope with the increase in data traffic given that deploying new infrastructure leads to important financial and environmental costs. In cellular networks, an attractive solution is to share resources between Mobile Network Operator (MNO), for example using roaming, radio access network sharing or radio access network slice [1]. More recently, marketplaces of connectivity have also been considered, involving financial transactions such as trading mechanisms or auctions whereas other solutions may rely on dynamic exchanges and fair cooperation between providers. In mobile cellular networks, two major use-cases for cooperation are generally considered: network coverage extension and network capacity extension. In the former case, some users may have no connectivity due to a lack of coverage of their home provider's infrastructure. In the latter case, there may be too many users with respect to the network cell capacity of their home provider. To overcome these problems, a provider may want to collaborate with one or more other providers to benefit from their resources resulting in closer or less loaded cells for the users in difficulty as in [2]. In some situations as in Figure 1, the providers may even envisage scenarios based on reciprocity and fair share of resources, so that each provider equally benefits from the collaboration.

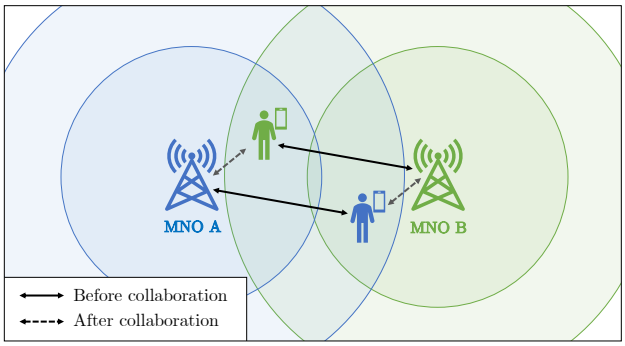


Fig. 1: Principle of multi-MNO collaboration: by mutually taking charge of their partner's user, each MNO decreases its cell load and optimises the connectivity.

To evaluate and identify a participation strategy in such collaborating scenarios, multi-agent learning offers compelling perspectives as it computes the optimal transactions between players. In particular, the recent emergence of Multi-Agent Reinforcement Learning (MARL) in non-cooperative games is very promising [3]–[5]. However, the study of Reinforcement Learning (RL) requires an environment for each specific task. To compensate the absence of such an environment for our use-case, we introduce a novel framework to create some fully adaptable RL environments for the simulation of multi-provider cooperation. We decided to adopt the OpenAI Gym toolkit which is a reference in the study of RL agents [6].

The paper is structured as follows. First, we explain in section II some useful concepts for a better understanding. Then in section III, we introduce our framework and describe in details its adaptable parameters. In section IV, we focus briefly on multi-agent interaction, in particular showing that this kind of cooperation falls into what is called a social dilemma. We conclude in section V in particular about further works.

II. Background

In this section, we explain some specific notions: Telecom vocabulary, Markov processes and the specifications of the Gym Framework.

A. Vocabulary of Telecom providers

1) MNO: In our framework, an agent (or player) represents a Telecom Provider called a Mobile Network Operator (MNO).

2) Sites: Each MNO owns a set of base stations also called cell sites. Moreover, we consider that each site is divided into a fixed number of sectors which delimit the so-called cells. We use the term cell for these sectors (see Figure 2). Usually, the number of cells is fixed to 3 but we will see later that it can be modified.

3) Users: In the environment, some users (or user-equipments) stand around the sites. Each user is a customer of one and only one provider (MNO) and is covered by the closest site owned by its provider.

4) Radio resources: For each site, and for each cell, the MNO can share a part of its resources.

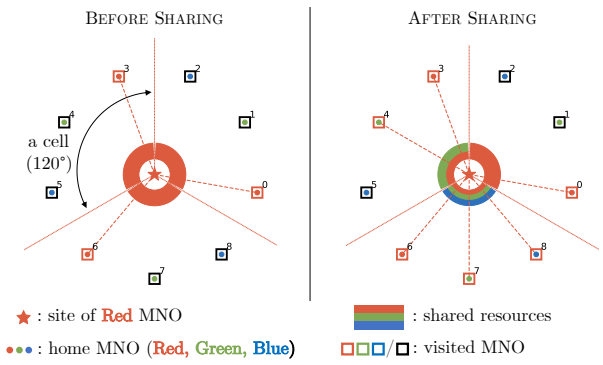


Fig. 2: Example of environment with three players (MNOs): Red, Green and Blue. There are one site (owned by *Red*) and nine users (three per MNO). The colour of a dot identifies the user's home MNO (i.e. the MNO to which the user is subscribed) ; the colour of a dotted line identifies the user's visited MNO (i.e. the MNO providing the radio resource)

B. Partially Observable Markov Decision Processes

In the literature, Reinforcement Learning environments are commonly formalised as Partially Observable Markov Decision Processes (POMDP) which are a generalisation of Markov Decision Processes (MDP) [7]. POMDP are usually defined as a 6-tuple (S, A, T, R, Ω, O) where:

- S is a set of states
- A is a set of actions
- $T : S \times A \to [0, 1]$ is a (stochastic) transition function
- $R : S \times A \to \mathbb{R}$ is a reward function
- Ω is a set of observations
- $O : S \times A \times \Omega \to [0, 1]$ is an (stochastic) observation function

The goal is to find a policy $\pi : s \mapsto a$ so that the cumulated reward R_t is maximised.

POMDP can be solved by exact methods like value-iteration algorithms when the model is known thanks to the Bellman equations. When the model is unknown and when the spaces are not too high, it is necessary to use RL algorithms such as the classical Q-Learning or REINFORCE [8]. Finally, when states and/or actions sets are very high or even continuous, an approximation function can be used, in particular using neural networks which made the emergence of Deep Reinforcement Learning [9].

C. Gym environments

OpenAI Gym is a toolkit [6] which allows hosting RL environments and interacting with them through Reinforcement Learning strategies. Therefore Gym is practical to evaluate and compare Reinforcement Learning agents in a generic way. A Gym environment is synthesised by four main application programming interface (API) functions:

1) `env.init()`: the environment is set with the initial set (which can be randomised)
2) `env.reset()`: the same than the previous function but after initialling at least once the environment
3) `env.step(state, action) = [new_state, reward, done, info]` : the most important function among the four, it allows to execute an action to move to another state. Doing a step, the agent receives its reward and an "end of episode" Boolean (`done`) and some optional information.
4) `env.render()`: allows to see a visual of current state of environment

III. Our Multi-agent framework

In this section, we present our Telecom Providers Cooperation environment[1].

To be integrated as an OpenAI Gym environment, and therefore benefit from Gym's flexibility, our environment implements the four API functions mentioned above. In addition, we also describe its configurable features as well as the major provided functions.

A. Environmental features

In our framework, some preset environments are available which are described in III-C but one can also create specific one in which we can modify the following parameters :

[1]The source code of our framework is available on GitHub: https://github.com/tlgleo/gym-MNOs-cooperation

1) Number of agents: `n_agents` : In the context, the number of agents (or players) is the number of MNOs.

2) Positions of sites: `positions_sites` is a list of `n_agents` lists containing the (x, y) fixed positions of sites

3) Number of cells: `n_cells` determines the number of cells a site can provide in a given frequency band. In most deployed networks, 3 is a typical value, but next generation networks like 5G networks may review this significantly.

4) Positions of users: User equipments can be considered fixed or mobile (see III-A5). In all cases, the number and initial positions of users have to be defined. (x, y) positions have to be defined in the list `positions_users`. A function called `random_users` is also available to define random positions. At last, each user has a fixed affiliation to one and only one player (provider). This is defined in the list `clients` containing the MNO identifier for each user.

5) Mobility of users: Users can be defined motionless or mobile. For the latter case, we created a class `Kinematics` which is used to define a profile of movement for users. The preset type called `'line'` is a simple line movement with constant speed[2].

6) Discretization of observation: A vector-type observation can be useful for some agent algorithms like those using Deep RL. Therefore an option of the environment allows to convert observation in multi-layer grid-world which fits to many Deep RL (such as CNN) algorithms (Also see III-B2)

B. Gym Features

In this section, we describe the POMDP features (section II-B) of our Gym toolkit (II-C).

1) Actions: At each step, each agent (MNO) plays an action which is the partition of its resources for each cell of each site. Formally, an action is an `numpy` array a of shape `(n_sites, n_cells, n_agents)` where:

$$\forall i_S,\ \forall k_C,\ \sum_{i_P=1}^{N} a[i_S, k_C, i_P] = 1 \tag{1}$$

with N the number of MNOs (players) and i_S, k_C, i_P respectively the indices of sites, cells and players.

For example, Figure 3 displays four examples of environment `env_3A_3S_9U-v2`, where the action performed by the first agent (Red agent) was respectively:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

[2]An example of mobility for the environment `env_3A_5S_30U-v0` can be viewed here : https://www.youtube.com/watch?v=ZfCgvPOmUoc

with the cell's index incrementing in anticlockwise direction beginning by the right one.

2) Observations: The observation state is subject to discussion. A multi-agent with partial observation is considered as non-Markovian which often requires to create a state with the history of actions. Such a state is created within algorithms [10], [11], that is why environments only need to provide current observation. We then propose a state available in two versions: total and partial observations. It provides a bitmap showing the positions of the sites and users. Such a bitmap is suited for most of Deep RL algorithms using Convolutional Neural Networks [9], [12]. In the figure 4, we show the two kinds of observations in an example.

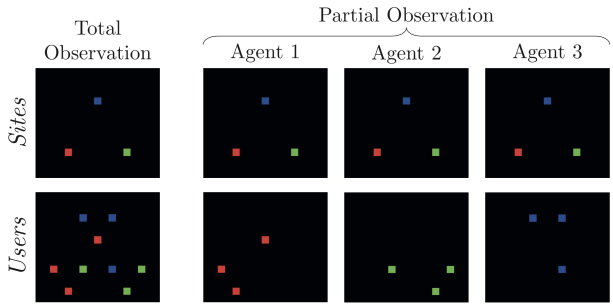


Fig. 4: Total and partial observations of example environment `env_3A_3S_9U-v1` (shown in Figure 3)

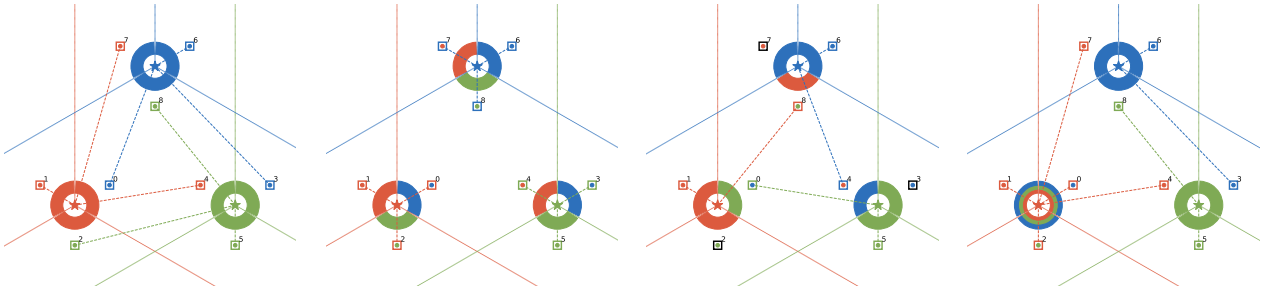
3) Transition function: The users are connected to sites/cells according to the shares that players allocate to each other. To compute the link, we decided to adopt a simple greedy algorithm. The links are decided in an increasing order of distances of all tuples user-sites (u, s) when the proper cell of s allocates enough resources to u's provider. Note that, although this allocation is not optimal, it is rather realistic to model such a cooperative environment. In addition, a more optimal solution would be NP-hard and an optimisation with good computational performance is out of the scope of the paper. Figure 3 shows links in the same environment with different allocations.

4) Rewards: For the reward of our Gym environment, we propose to use a reward (or utility) function R_P for each player P (provider) inspired from [13].

$$R_P = \sum_{u \in \mathcal{C}_P} \frac{\Gamma_u}{m_u} \exp(-\frac{D_u}{D_{max}}) \tag{2}$$

where :

- $\mathcal{C}_P$ is the set of customers of P
- Γ_u is the share of radio resources allocated for provider of u (P) in the cell
- m_u is the number of customers of P standing in the cell of u
- D_u is the distance between u and its connecting antenna, D_{max} is a scaling constant



(a) Initial state (b) Optimal situation (c) Bad situation (d) Only one cooperator

Fig. 3: Four instances of a simple `env_3A_3S_9U` environment (3 Red/Green/Blue players each owning 1 site and 3 users). Fig 5b is one optimal cooperation. Fig 5d shows a situation where only one player (red one) agree to cooperate

C. Examples of predefined environments

Our environment can be instantiated with any combination of players (MNOs), sites and users. For example, Figure 5 shows four simple canonical environments with two or three players, one or two sites per player and one to ten users. These environments are directly available in the framework.

IV. Game Theory analysis

Though implementing agents is out of the scope of the paper, we found interesting to focus on Game Theory aspects. After explaining some definitions, we will show that multi-player games in our environments can be considered as a social dilemmas.

A. Game Theory and Stochastic Games

Multi-agent interaction leads to game theory issues of which we explain here some basics.

1) Strategy and Utility: A strategy (also called policy) is basically a function that maps personal states ($s^{(i)} \in S$) to actions ($a^{(i)} \in A$) : $\pi_i : s^{(i)} \mapsto a^{(i)}$. In a game theory approach, each agent (or player) is driven by a utility function which measures the satisfaction of a strategy. A given strategy can be based on the gain, money, popularity, number of points, quality of service (for telecom), etc.

2) Nash Equilibrium: A major notion of Game Theory is the Nash Equilibrium which can be simplified as a situation where every agent has no incentive to change alone its strategy. Formally, the Nash Equilibrium is defined as follows: if $G^{(i)}(\pi_i)$ is the payoff of agent A_i, a joint strategy $(\pi_i^*)_{i\in\mathcal{I}}$ is said to be a Nash Equilibrium if [14]:

$$\begin{aligned} &\forall i \in \mathcal{I},\ \forall \pi_i,\ G^{(i)}(\pi_i^*, \pi_{-i}^*) \geq G^{(i)}(\pi_i, \pi_{-i}^*) \\ &\text{with } \pi_{-i} = [\pi_0, ..., \pi_{i-1}, \pi_{i+1}, ..., \pi_{N-1}] \end{aligned} \tag{3}$$

3) Stochastic games: Stochastic games are an extension of POMDP (II-B) with multiple decision makers [15]. N agents $I = \{1, ..., N\}$ play in the same POMDP game where S and A are extended to product sets $\times_{i\in I} S_i$ and $\times_{i\in I} A_i$. Each agent $i \in I$ receives a distinct reward $r^{(i)}$ that it wants to independently maximise.

B. Social Dilemmas

As seen in previous works [16] for a similar use-case, we show here that the games we consider in the paper can be considered as sequential social dilemmas [17]. We explain in this section its principle. Let us consider the case with $N = 2$ players (providers) and with simply two possible actions: cooperate and defect. The figure 6 shows the four possible payoffs.

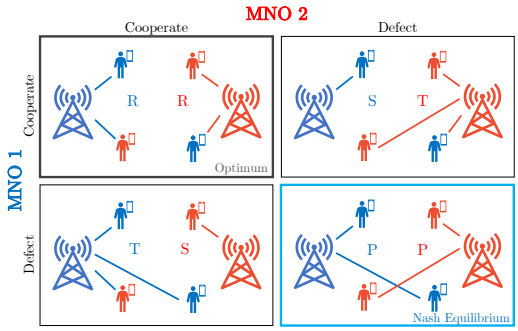
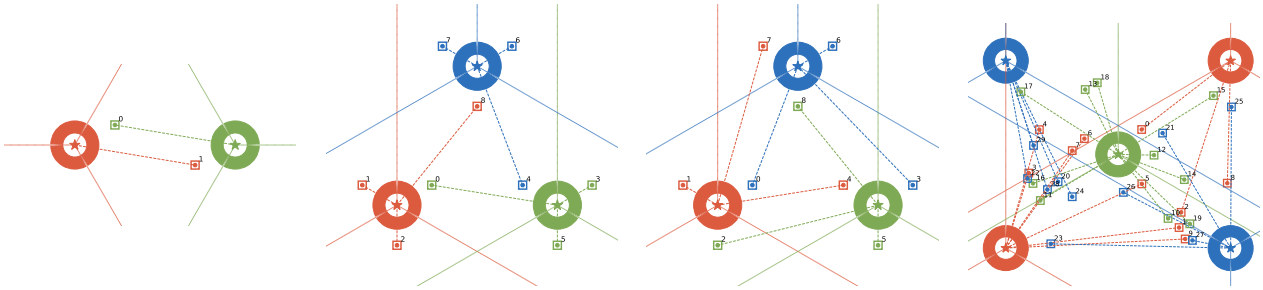


Fig. 6: A two-player stochastic game in situation of Sequential Social Dilemma : symmetric payoffs verify $S < P < R < T$. Mutual defection is the Nash Equilibrium while the optimal outcome is the mutual cooperation. For further details, see [17]

Such a game is called social dilemma if the following inequalities are verified:

- $R > P$ (1)
- $R > S$ (2)
- at least one of these two inequalities:
  - $T > R$: greed (3a)
  - $P > S$: fear (3b)

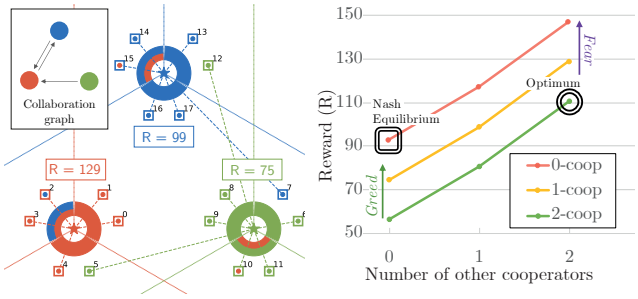
(1) means that mutual cooperation is better than mutual defection and (2) that mutual cooperation is better than being exploited. Formally, a social dilemma is a game which admits at least one non-optimal Nash equilibrium in particular (*Defect, Defect*) in Prisoner's Dilemma (where greed (3a) and fear (3b) are verified).



(a) `env_2A_2S_2U-v0` (b) `env_3A_3S_9U-v1` (c) `env_3A_3S_9U-v2` (d) `env_3A_5S_30U-v0`

Fig. 5: Some examples of environments already available : a very simple two-players prisoner dilemma case (a), some three-players ones (b) and (c) with circular and reciprocal cooperation, and finally a random three-players one (d)

Social dilemmas exist with more than two players. In the figure 7b, we represent a three-player game modelled by the environment `env_3A_3S_18U-v0` (an available predefined environment) with the reward function detailed in III-B4. We emphasis the dilemma with a Schelling diagram since payoffs tables are illegible with more than two players.



(a) `env_3A_3S_18U-v0` (b) Schelling Diagram

Fig. 7: A three-players social dilemma shown by a Schelling Diagram (for a better understanding, see [17] or [18]). *Red* and *Blue* are cooperating with each other and *Green* cooperates only with *Red*.

V. Conclusion

Providers may consider cooperating to meet exploding connectivity needs. To do so, they might adopt a simple game theory approach without financial transactions where players (providers) make transactions so that they all increase their quality of service. These assumptions then lead among other things to the study of agents behaviour in a context of telecom resources sharing. To anticipate the lack of some realistic environment where agents can learn to optimally interact, we implemented a framework able to generate some customizable RL environments in the OpenAI Gym toolkit. The main goal of this article is to introduce the provided features of our framework. We additionally presented a brief study of game theory situation of the problem. We showed in particular that our considered exchange example can be view as a so-called social dilemma, a situation where players have no incentive to collaborate alone despite the optimal mutual cooperation. In further works, we plan to generate environments derived from our framework to study algorithms known to be robust for multi-players social dilemmas.

References

[1] Annie Turner. Inception: Digital twins for 5g network infrastructure-sharing, Oct 2020.

[2] Ricardo Tavares and Martyn Roetter. Beyond mobile-network sharing: Regulatory challenges in dense urban areas. 2020.

[3] Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. 2017.

[4] Julien Perolat, Joel Z Leibo, Vinicius Zambaldi, , et al. A multi-agent reinforcement learning model of common-pool resource appropriation. In *Advances in Neural Information Processing Systems*, pages 3643–3652, 2017.

[5] Ryan Lowe, Yi I Wu, Aviv Tamar, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pages 6379–6390, 2017.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, et al. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[7] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

[8] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[9] Volodymyr Mnih, Kavukcuoglu, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[10] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *arXiv preprint arXiv:1507.06527*, 2015.

[11] Long-Ji Lin and Tom M Mitchell. *Memory approaches to reinforcement learning in non-Markovian domains*. Citeseer, 1992.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, et al. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[13] Mohammad T Kawser, Nafiz Imtiaz Bin Hamid, Md Nayeemul Hasan, et al. Downlink snr to cqi mapping for different multipleantenna techniques in lte. *International journal of information and electronics engineering*, 2(5):757, 2012.

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

[15] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[16] Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, and Benoît Radier. Game theory approach in multi-agent resources sharing. In *25th IEEE Symposium on Computers and Communications (ISCC)*, 2020.

[17] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, et al. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 464–473, 2017.

[18] Thomas C Schelling. *Micromotives and macrobehavior*. WW Norton & Company, 2006.