Deep Reinforcement Learning for Controlled Piecewise Deterministic Markov Process in Cancer Treatment Follow-up

A. Cleynen[1,2], B. de Saporta[1], O. Rossini[1], R. Sabbadin[3], M. Vinyals[3]

[1] IMAG, Univ Montpellier, CNRS, Montpellier, France
[2] John Curtin School of Medical Research, Australian National University, Canberra, ACT, Australia
[3] Univ Toulouse, INRAE-MIAT, Toulouse, France

orlane.rossini@umontpellier.fr

Résumé

Le myélome nécessite un suivi à long terme et se caractérise par des phases de rémission et de rechute, pendant lesquelles un marqueur est surveillé et sert de base à une politique de traitement. Nous modélisons la dynamique du marqueur par un Processus Markovien Déterministe par Morceaux contrôlé à observations bruitées, espace d'état continu et à modèle partiellement connu et nous proposons une nouvelle méthode de contrôle pour ce PDMP. Nous transformons ce problème en Processus de Décision Markovien Partiellement Observé à espace d'état continu, sur lequel nous mettons en oeuvre un algorithme d'apprentissage par renforcement profond. Nous montrons expérimentalement sur des trajectoires de marqueur simulées que cet algorithme permet une prise de décision efficace.

Mots-clés

Processus Markovien Déterministe par Morceaux, Processus de Décision Markovien Partiellement Observé, Contrôle Stochastique, Apprentissage par Renforcement Profond, Optimisation de Traitement.

Abstract

Myeloma requires long-term follow-up and is characterized by phases of remission and relapse, during which a marker is monitored and serves as the basis for a treatment policy. We model the dynamics of the marker by a controlled Piecewise Deterministic Markov Process with noisy observations, continuous state space and a partially known model and we propose a new control method for this PDMP. We suggest a transformation of this problem into a continuous state space Partially Observable Markov Decision Process, on which we implement a deep reinforcement learning algorithm. We show experimentally on simulated marker trajectories that this algorithm allows effective decision-making.

Keywords

Piecewise Deterministic Markov Process, Partially Observable Markov Decision Process, Stochastic Control, Deep Reinforcement Learning, Treatment Optimisation.

1 Introduction

Patients suffering from long-term illnesses such as myeloma alternate between phases of remission and relapse. Their follow-up includes regular blood tests to measure a specific biomarker, among other variables. Although several variables are monitored, in this paper, we focus on simplified treatment decisions based solely on this specific biomarker. These indirect and noisy measurements often lack detailed knowledge of the dynamics of the process, which vary from patient to patient. While medical steps facilitate follow-up decisions, personalized criteria still need to be established for better personalised patient management. This requires a precise understanding of disease dynamics and reliable predictive algorithms. It is crucial to develop a model capable of representing the biomarker dynamics for all patients, while also being adaptable with patient-specific parameters. Additionally, online adaptive models, which adjust treatment as each new data is observed, are essential for relapse prediction and treatment automation. Piecewise Deterministic Markov Processes (PDMPs) are non-diffusive hybrid stochastic processes, that handle continuous and discrete variables in continuous time. PDMPs are characterized by deterministic motion punctuated by random jumps at random times, making them suitable for modelling systems where change occurs both continuously and at discrete points in time. They are thus simple to simulate and easy to interpret [7]. Controlled PDMPs extend the concept of PDMPs by introducing the capability for decision-making in continuous time. This means that in controlled PDMPs, an external controller influences the system dynamics by making decisions that affect both the deterministic paths and the timing or nature of the jumps.

Our focus centres on the computational resolution of a specific category of impulse control problems for PDMP. Impulse control for PDMPs involves selecting actions and intervention dates [6]. Finding the optimal treatment decision in continuous-time and continuous-state impulse control problems, especially when the process is only partially observed, presents a significant challenge. It is even more challenging when jump times remain hidden and the underlying model is partially unknown and can only be simulated. Previous approaches [3, 4], propose to formulate the controlled PDMP problem as a Partially Observable Markov Decision Process (POMDP). Then they resort to discretizing the belief state space of the POMDP and employing dynamic programming to approximate the value function, in order to address continuous state space and partial observability. While effective (even though computationally intensive), this method is constrained by its reliance on explicit model knowledge and the discretization process. Approximation algorithms were developed for POMDP with discrete state space and an unknown model (e.g. [2]) but these approaches generally require some knowledge of a good policy class, as well as substantial amounts of data. An alternative strategy [5, 12] adopts a simulation-based approach based on Monte-Carlo planning, that also handles continuous state space.

In this paper, we propose a new solution approach based on Deep Reinforcement Learning (DRL) [13]. While offering generalization capabilities, our approach allows to learn a control policy by interacting with a simulator of the model. As in previous work, we transform the controlled continuous-time PDMP problem into a discrete-time POMDP. Then, given the complex observation space of this discrete-time POMDP, composed of both discrete and continuous variables, we employ the Deep-Q-Network algorithm (DQN) [11, 9, 14] to solve it, thereby avoiding any discretization of the observation space.

2 Cancer treatment follow-up models

In our medical scenario, a patient enrols in a clinical trial for a time H at the onset of a remission phase. Throughout remission, the biomarker hovers at the nominal threshold ζ_0. In the absence of treatment, a relapse triggers an exponential surge in the biomarker level, culminating in the patient's death upon reaching the critical value of D. Treatment interventions succeed in lowering the biomarker level, yet with each relapse, the probability of treatment resistance escalates. This intricate interplay involving phases of remission, relapse, and treatment response constitutes the fundamental essence of our impulse control problem. Our investigation starts with delineating a specialized class of impulse control problems designed for Piecewise Deterministic Markov processes (PDMPs), a model initially proposed in [5]. Specifically, we consider the scenario where the spread of cancer relapse is treated as unknown, adding a layer of complexity to the problem. We describe the translation of our control problem into a Partially Observable Markov Decision Process (POMDP) framework.

2.1 Piecewise Deterministic Markov Process

We consider an impulse control problem for hidden piecewise deterministic Markov processes (PDMPs). The mode (m, k) corresponds to the patient's overall state of health ($m = 0$: remission, $m = 1$: relapse, $m = 2$: non-curable relapse, $m = 3$: death) and $k \in \mathbb{N}$ (the number of curable relapses). The biological marker level is denoted by $\zeta \in [\zeta_0, D]$ with ζ_0 the nominal value and D the death level and $u \in [0, H]$ is the sojourn time in a health' state (added for technical reasons to deal with semi-Markov condition), where H corresponds to the end of the patient's follow-up. Let the state space E be $E \subset \{0, 1, 2, 3\} \times \mathbb{N} \times [\zeta_0, D] \times [0, H]$. The complete state of the patient is denoted by $x = (m, k, \zeta, u)$ in E the state space.

Decisions are made throughout a patient's trajectory. Let $\mathbb{D}$ be the space of decisions such that $\mathbb{D} = \mathcal{L} \times \mathcal{R} \cup \{\Delta\}$. Control is expressed as a decision pair : $d = (\ell, r)$, where $r \in \mathcal{R} = \{15, 30, 60\}$ is the delay (prescribed by the doctor) until the next visit. Visits correspond to the biomarker level measurement and the adjustment of the treatment according to the results. The therapeutic choice is $\ell \in \mathcal{L} = \{\emptyset, a, b\}$ ($\ell = \emptyset$: *no treatment*, $\ell = a$: *chemotherapy* and $\ell = b$: *palliative care*). The decision $d = \Delta$ corresponds to the action *do nothing* and applies only when the patient is dead.

A PDMP on the state space E is defined by three local characteristics $(\mathbf{\Phi}, \lambda, \mathcal{Q})$. The flow $\mathbf{\Phi}$ describes the deterministic trajectory of the process between jumps. It depends on the control applied and in particular on the treatment : $\mathbf{\Phi}^\ell(x, t) = (m, k, \Phi^\ell_{m,k}(\zeta, t), u + t)$, where :

$$\Phi^\ell_{m,k}(\zeta, t) = \zeta e^{v^\ell_{m,k} t}$$

describes only the trajectory of the biological marker between jumps. When the patient is dead, no treatment is applied and the flow is $\mathbf{\Phi}^\Delta(x, t) = (3)$. The biomarker evolution depends on the therapy choice, the disease regimen and the number of relapses.

Let $t^{\ell\star}(x)$ be the deterministic time the flow takes to reach the boundary of the state space E. Let $\partial E = \{1, 2\} \times \mathbb{N} \times \{\zeta_0, D\} \times (0, H]$ be the boundary on E. The time $t^{\ell\star}(x)$ also depends on the treatment and the disease regimen :

$$t^{\ell\star}_{m,k}(\zeta) = \inf\{t > 0 : \Phi^\ell_{m,k}(\zeta, t) \in \partial E\}$$

In PDMP, a jump refers to a sudden and instantaneous change in the state of the system. The jump intensity λ quantifies the frequency at which these jumps occur. Treatment also influences the risk function $\lambda^\ell(x) = \lambda^\ell_{m,k}(\zeta, u)$. Notably, there are two distinctive types of relapse scenarios considered : standard relapses occurring during remission phases and relapses indicative of therapeutic escape. For standard relapses, the probability of occurrence increases with the duration of time spent in remission. On the other hand, the risk of relapses associated with therapeutic escape is influenced by the biomarker level. In light of these considerations, we choose a Weibull distribution of the form : $f_{0\to m'}(u) = (\alpha_{m'} u)^{\beta_{m'}}$ and $f_{1\to 2}(\zeta) = (\alpha' \zeta)^{\beta'}$.

The Markov kernel Q provides a probabilistic mapping from the pre-jump state to the post-jump state. In remission, the patient may transition to either a curable relapse in the absence of chemotherapy or to an incurable relapse. Incurable relapse occurs when cancer cells become resistant to chemotherapy. In the case of relapse and without treatment, the biomarker increases to the critical value D, leading to the patient's death. When chemotherapy is administered, the biomarker decreases to ζ_0 and returns to remission. Regardless of the treatment chosen, therapeutic escape may occur at any time. In the case of therapeutic escape, the biomarker increases, regardless of the administered treatment, toward the D threshold, ultimately resulting in the patient's death. When patients are dead, no jumps occur anymore.

Let $\mathcal{P}(x, d)(x')$ be the transition kernel associated with the continuous-time PDMP. The transition kernel of the PDMP combines the deterministic flow, the jump intensity and the Markov kernel. However, due to its extensive nature, detailed analytic formulas will not be included in this paper, but it is worth noting that they allow the kernel to be simulated easily. After initialization in a known initial state $x_t = (0, 0, \zeta_0, 0) \in E$ at time $t = 0$, the simulation of the PDMP proceeds as follows. At each decision point, an agent selects a decision $d = (\ell, r)$. The next visit point x_{t+r} is then simulated based on the selected treatment ℓ. This simulation involves updating the biomarker ζ according to the deterministic flow, checking if any jump occurs before the next visit and, if so, simulating a post-jump location according to the Markov kernel. This process iterates until the simulation horizon is reached.

2.2 Partially Observable Markov Decision Process

The trajectory of the process defined above depends on the sequence of decisions and the dates on which the decisions are made. We assume that visits take place at discrete dates $n_0 = 0, n_1, ..., n_k$, where the time lapse between two visits can be 15, 30 or 60 days. At most $N = \frac{H}{15}$ visits can occur. Moreover, decision-related constraints appear. The last visit must take place at the end H of the follow-up [1]. The variable $t \in [0, H]$ indicates the time elapsed since the start of the trajectory. In addition, treatment must be applied for a minimum of 45 days [1]. The variable $\tau \in [0, H]$ corresponds to the time since treatment (chemotherapy or palliative care) was administered. It can be shown that the impulse control problem described above can be formalized as a discrete-time partially observed Markov decision process (POMDP).

A POMDP is a tuple $(\mathbb{S}, \Omega, \mathbb{D}, \mathbb{K}, \mathcal{T}, C)$.

The state space $\mathbb{S}$ corresponds to the hidden state of a patient $s = (m, k, \zeta, u, t, \tau)$ in $\mathbb{S} \subset E \times [0, H]^2 \cup \{3\}$.

Blood measurements are intrinsically subject to variations independent of the medical condition. These fluctuations can be attributed to measurement errors, natural variations, and external influences. The biomarker is thus observed through a multiplicative noise as the biomarker is growing exponentially. Let $y = \zeta e^{\epsilon}$ with $\epsilon \sim \mathcal{N}(0, 1)$ be the noisy biomarker. In addition, the patient's overall health is not observed, except when the patient is deceased. Let $z = \mathbb{1}_{(m=3)}$ be the death indicator. At a given time t, the observation of a patient's condition is $\omega = (\tau, t, y, z)$ with $\omega \in \Omega$. The observation space is $\Omega \subset [0, H]^2 \times \mathbb{R}_+ \times [0, H] \times \{1\}$.

The decision space $\mathbb{D}$ remains unchanged.

$\mathbb{K}(\omega) \subseteq \mathbb{D}$ is the space of admissible decisions in observation ω. It is used to specify all allowed actions state by state : $\mathbb{K}(\omega) = \{d \in \mathbb{D}; (\omega, d) \in \mathbb{K}\} \neq \emptyset$. Constraints are only defined by observations.

$$\mathbb{K}(\omega) = \begin{cases} \{\Delta\} & \text{if } z = 1 \text{ or } t = H \\ (\ell, r) \in \{a, b\} \times \mathcal{R} & \text{if } \tau \in (0, 45) \text{ and } t + r \leq H \\ (\ell, r) \in \mathcal{L} \times \mathcal{R} & \text{such that } t + r \leq H \end{cases}$$

The POMDP joint transition-observation function of a state-observation tuple $(s, \omega) \in \mathbb{S} \times \Omega$ to state-observation tuple $(s', \omega') \in \mathbb{S} \times \Omega$ when action $d \in \mathbb{K}(\omega)$ is taken is denoted by $\mathcal{T}(s, \omega, d)(s', \omega')$. It can be expressed as a function of $\mathcal{P}(x, d)(x')$ the piecewise deterministic Markov process (PDMP) transition kernel. This means it can be written as a combination of PDMP flow, jump intensity and Markov kernel. Detailed analytic formulas are omitted here, but notice that the POMDP joint transition-observation function is set according to the PDMP parameters. It's worth mentioning that even if these parameters are unknown, they can be simulated using various techniques, based on real data, for instance. Therefore, the POMDP transition-observation function can be constructed based on simulated data, enabling the application of reinforcement learning techniques despite the uncertainty surrounding the true parameters.

Let C be the non-negative cost-per-stage function such that $C : \mathbb{D} \times \mathbb{S} \to \mathbb{R}_+$. In POMDPs, the cost function quantifies the cost associated with different decisions per stage. Cost function details are provided in section 4.2.

A history is a sequence of observations and decisions $h_n = \{\omega_0, d_0, \omega_1, \cdots, \omega_n\}$ and $\mathcal{H}_n$ is the set of histories of size n. Along a trajectory, the agent applies decision rules which map a history to an appropriate decision. Let $f_n : \mathcal{H}_n \to \mathbb{K}$ be a decision rule for the nth visit such that for all h_n in $\mathcal{H}_n$ we have $f_n(h_n)$ in $\mathbb{K}(\omega_n)$. We define an admissible policy π as a sequence of decision rules $\pi = (f_n)_{0:N-1}$ and Π the set of all admissible policies. Then, the total cumulated cost from visit n is defined as follows $C_n = \sum_{k=n}^{N-1} C(D_k, S_{k+1})$.

The value function $V^{\pi}(h_n) = \mathbb{E}_{\pi}[C_n | h = h_n]$ is the expected total cumulated cost, starting from history $h_n \in \mathcal{H}_n$ when following policy π. Our next objective is to obtain an optimal policy $\pi^{\star}$ such that the value function V is optimal : $V^{\star}(h) = \min_{\pi \in \Pi} V^{\pi}(h)$ for all $h \in \mathcal{H}$.

Since we assume the existence of a trajectory simulator, the application of Reinforcement Learning (RL) methods emerges as a sensible choice. However, considering the complexity of the state space (hybrid discrete and continuous) and the partial observability of the state, turning to deep RL strategies seems compulsory.

3 Deep reinforcement learning solution Strategies

Reinforcement Learning (RL) methodologies can be broadly categorized into two principal families : value learning and policy learning approaches. These approaches diverge in their strategies for addressing sequential decision

[1] We assume H is a multiple of 15.

problems. Value learning focuses on assessing and enhancing the value function associated with a given policy, aiming to identify the optimal value for each state. On the other hand, policy learning directly updates the policy, determining the optimal sequence of actions for each state. Notably, policy learning often achieves faster convergence. However it generally results in stochastic policies, while value-based approaches ensure a deterministic policy. This is a crucial requirement in cancer monitoring and treatment, since doctors and patients are reluctant to apply non-deterministic treatment strategies.

Deep Q-Network (DQN) [11] represents a state-of-the-art approach for handling Markov Decision Processes (MDPs) with continuous state spaces. While DQN can also be applied to Partially Observable Markov Decision Processes (POMDPs), its inability to handle historical data may limit its performance (see concluding remarks). However, this is a good starting point approach, which we adopt here.

4 Experimental evaluation of DQN for cancer treatment

4.1 Implementation details

For the evaluation, we implemented a Gymnasium[2] compatible RL environment in Python to simulate the trajectories of the cancer treatment follow-up POMDP model (presented in Section 2.2). For the DQN algorithm, we used the implementation available in RLlib[3], an open-source Python library specialized in the evaluation of Deep Reinforcement Learning (DRL) algorithms. Our experiments code and data are available in a GitLab repository[4].

4.2 The cost function

Our objective is to choose the best available treatment and the best next-visit date to minimize the long-term impact on the patient's quality of life. This is achieved through designing a cost function encoding the diverse short-term impacts of each treatment. Designing such a cost function is in itself a difficult task. Recall that $s = (m, k, \zeta, u, t, \tau)$ and let $s' = (m', k', \zeta', u', t', \tau')$ and $d = (\ell, r)$. We propose to use the following cost function definition :

$$C(s, d, s') = C_V + \kappa|\zeta' - \zeta_0|r + \beta r \mathbb{1}_{\zeta=\zeta_0, \ell\neq\emptyset} + C_D \mathbb{1}_{\zeta'=D}$$

as defined in [5], where C_V is a visit cost, κ is a non-negative scale factor penalizing high marker values, β is a penalty for applying an unnecessary treatment, r is the treatment duration and C_D is the death cost.

In order to handle action constraints implicitly in DQN, we extend the cost function, so that it returns large cost values for invalid actions, as follows :

$$C_K(s, d, s') = C(s, d, s') + L_H \mathbb{1}_{t'>H} + T_F \mathbb{1}_{\ell=\emptyset, 0<\tau<45}$$

where L_H is a cost for exceeding the horizon and T_F penalizes stopping treatment too early. While calibrating cost parameters C_V, κ, β, and C_D is a difficult task, L_H and T_F parameters were simply set to arbitrary large values to prevent forbidden actions.

Policy	**Mean Cost**
Naive	55509.49 ± 1931.78
Threshold	5300.17 ± 173.31
Inactive	843.12 ± 101.92
DQN	978.17 ± 137.29

TABLE 1 – Policy evaluation performance on simulations

4.3 Patient follow-up optimization

We applied the Deep Q-Network (DQN) algorithm and three heuristic control approaches to the patient problem.

We evaluated the performance of each algorithm by the average total cost incurred over 10^5 Monte-Carlo simulations. To provide a performance comparison, we evaluated the performance of three other arbitrary policies.

Naive policy consists of selecting decisions randomly (corresponding to our upper bound cost).

Threshold policy operates according to two last observations. If the estimated biomarker level falls below 5, no treatment is applied, and the next visit is scheduled in 30 days. If the biomarker level is estimated to be between 5 and 25, chemotherapy is administered, and the next visit is planned in 30 days too. If the biomarker level is estimated to be above 25, palliative care is administered, and the next visit is scheduled in 60 days.

Inactive policy consists of not administering any treatment and scheduling a visit every 60 days.

Table 1 presents the performance evaluation of the different policies on simulations. It is worth noting that the Inactive policy serves as the lower bound cost and the Naive policy serves as the upper bound cost for comparison purposes. The DQN policy, which is the focus of the evaluation, shows a mean cost of 978.17 with a confidence interval spanning $[840.88, 1115.46]$. Interestingly, there is an overlap between the confidence interval of the DQN policy and that of the Inactive policy. This suggests that while the DQN policy performs worse on average compared to the Inactive policy, it still achieves comparable results within a certain confidence range. These findings underscore the potential of deep RL methods in decision-making processes for patient follow-up. However, it is crucial to note that our cost function needs to be parameterized differently, as we do not intend for the Inactive policy, which refrains from administering any treatment, to represent our optimal policy.

5 Conclusion

In conclusion, monitoring and treatment of myeloma can be modelled by a hidden controlled Piecewise Deterministic Markov Process (PDMP). We reduce the problem into an equivalent discrete-time Partially Observable Markov Deci-

[2] https://gymnasium.farama.org/index.html

[3] https://docs.ray.io/en/latest/rllib/index.html

[4] https://forgemia.inra.fr/orlane.le-quellennec/controlled_pdmp_po

sion Process (POMDP). We propose to employ deep Reinforcement Learning (DRL) to learn control policies for such POMDP. DRL offers the advantage of not necessitating an explicit model, only an environment capable of simulating trajectories. Furthermore, by employing deep neural networks as function approximators, deep RL algorithms can directly handle the complex observation space of the cancer follow-up problem without any discretization. Finally, despite the partial observability inherent in POMDPs may require policies that leverage the entire history (i.e., past actions/observations) to be effective, the memoryless Deep Q-Network (DQN) algorithm, seems effective (especially compared to a threshold policy).

Of course, our experiments are really preliminary and should be extended. Additionally, while the DQN policy performs well in terms of cost, its decision-making process lacks interpretability, necessitating additional investigation. The subsequent phase of our research aims to compare the performance of the memory-less DQN algorithm with that of the Recurrent Replay Distributed DQN (R2D2) [10] algorithm, an extension that considers histories by combining DQN with a Long Short-Term Memory (LSTM). We conjecture that this paradigm shift will yield improved decision-making policies for cancer treatment, to the price of a more complex representation of treatment policies (not based on the current observation but potentially on a fill history of treatment/observations).

The exploration of alternative modelling avenues remains a compelling direction for future research. Incorporating model knowledge and moving to a belief state formulated MDP could be an interesting avenue. We hypothesize that a more informative framework will lead to more efficient decision-making. Consequently, leveraging model-based RL methods, particularly Bayesian RL [8], holds promise in learning the underlying model and policy more effectively. By integrating prior knowledge about the system dynamics and uncertainties into the learning process, Bayesian RL approaches can provide more accurate predictions and better decision-making capabilities. This avenue of research could significantly enhance our understanding of complex systems like the one described, paving the way for more robust and efficient control strategies in the future.

Acknowledgements

We acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-21-CE40005 (project HSMM-INCA), of European Union's Horizon 2020 research and innovation program (Marie Sklodowska-Curie grant agreement No 890462) and the support of MESO@LR-Platform at the University of Montpellier.

Références

[1] Myeloma : diagnosis and management. *National Institute of Health and Care Excellence (NICE)*, 2016.

[2] J. Baxter and P. L. Bartlett. Infinite-Horizon Policy-Gradient Estimation. *Journal of Artificial Intelligence Research*, 15 :319–350, November 2001.

[3] Alice Cleynen and Benoîte de Saporta. Change-point detection for piecewise deterministic Markov processes. *Automatica*, 97 :234–247, November 2018.

[4] Alice Cleynen and Benoîte de Saporta. Numerical method to solve impulse control problems for partially observed piecewise deterministic Markov processes. *arXiv*, 2023.

[5] Alice Cleynen, Benoîte de Saporta, Aymar Thierry D'Argenlieu, and Régis Sabbadin. Medical follow-up optimization : A Monte-Carlo planning strategy. *arXiv*, 2024.

[6] O. L. V. Costa and M. H. A. Davis. Impulse control of piecewise-deterministic processes. *Mathematics of Control, Signals, and Systems*, 2(3) :187–206, 1989.

[7] Mark H. A. Davis. Piecewise deterministic markov processes : A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society Series B (Methodological)*, 46 :353–376, 1984.

[8] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian Reinforcement Learning : A Survey. *Foundations and Trends® in Machine Learning*, 8(5-6) :359–483, November 2015. Publisher : Now Publishers, Inc.

[9] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-Learning. In *AAAI'16 : Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094–2100. 2016.

[10] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent Experience Replay in Distributed Reinforcement Learning. *International conference on learning representations*, September 2018.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. 2013.

[12] David Silver and Joel Veness. Monte-Carlo Planning in Large POMDPs. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[13] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

[14] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. In *ICML'16 : Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1995–2003. JMLR.org, June 2016.