Reinforcement Learning for Optimal Chemotherapy Delivery

Jean-Guillaume Brasier, Amine Abdeljaoued

6.231 - Dynamic Programming and Reinforcement Learning
Massachusetts Institute of Technology
May 9, 2022

1 Introduction

Cancer is a disease in which some of the body's cells grow uncontrollably and spread to other parts of the body. These cells may form tumors, which are lumps of tissue that can be cancerous or not (benign). Cancerous tumors spread into, or invade, nearby tissues and can travel to distant places in the body to form new tumors (a process called metastasis) [10]. Being able to diagnose at an early stage and having a personalized treatment are effective at treating this disease and significantly increase survival chances.

Decision support systems based on reinforcement learning (RL) have been implemented to facilitate the delivery of personalized care. RL has great potential for enhancing decision making in critical care [3]. Our study stems from 2 important papers [7] [11] where a reinforcement learning framework was applied to cancer chemotherapy treatment.

The motivation is that in chemotherapy treatments, drug dosing through time is often done in a subjective manner, depending on the hospital and the doctors. Furthermore, there is a trade-off between administering a heavy chemotherapy treatment but risking long term side-effects, and administering a lighter treatment with the possibility of not fully curing the cancer. In the case of dynamic treatment regimes as stated above, reinforcement learning can be a powerful tool in simulating chemotherapy trials for drug development.

One of the main challenges is modelling the complex nonlinear dynamic systems associated with cancer development. Simple logistic and mathematical models are not able to account for all the variations in patient dynamics. We use model-free Deep Reinforcement Learning (DRL) to design a personalized cancer therapy schedule depending on two models of increasing complexity. The first model is based on a macroscopic view of cancer progression and overall patient condition, while the second model is based on cell dynamics. We will use various state-of-the-art reinforcement learning algorithms and compare their performances on these two models.

2 Mathematical Formulations

2.1 Patient-condition model [11]

This model is a first approach in capturing the effect of drug dosage on the wellness of the patient and the overall tumor growth as defined in [11]:

$$\dot{W}_t = a_1(M_t \vee M_0) + b_1(D_t - d_1)$$

$$\dot{M}_t = [a_2(W_t \vee W_0) - b_2(D_t - d_2)] \times 1\{M_t > 0\}$$

The states are $\mathbf{x_t} = (W_t, M_t)$ where W_t is the negative part of wellness (in response to chemotherapy and toxicity) - and M_t the tumor size at time t. The actions are D_t, the

chemotherapy agent dose levels that range from 0 to 1. The time horizon is $T = 12$ months, and updates happen at each month. The reward function is a combination of three functions that take into account the survival status, the wellness effects, and the tumor size effects (see Appendix A). For the survival function, we decided to give more importance to the wellness W because it has not been considered for a long time, with most studies neglecting the well-being of patients in favor of tumor decrease. M_0 and W_0 are independently generated from the uniform distribution on $(0, 2)$, D_0 on $(0.5, 1)$ and further drug doses (actions) on $(0, 1)$. We let $a_1 = 0.1$, $a_2 = 0.15$, $b_1 = 1.2$, $b_2 = 1.2$, $d_1 = 0.5$ and $d_2 = 0.5$. Termination occurs when the survival function returns death, when the tumor size reaches 0 or when the 12 time-steps have occurred.

2.2 Cell-specific model [7]

A more realistic model should include tumor growth, the immune system's reaction, and the effects of chemotherapy treatment on cells. The following system of ODEs as defined in [7] simulate patient's response to treatment through a pharmacological model of cancer chemotherapy.

$$\begin{aligned}
\dot{N}_t &= r_2 N_t \left(1 - b_2 N_t\right) - c_4 N_t T_t - a_3 N_t C_t \\
\dot{T}_t &= r_1 T_t \left(1 - b_1 T_t\right) - c_2 I_t T_t - c_3 T_t N_t - a_2 T_t C_t \\
\dot{I}_t &= s + \frac{\rho I_t T_t}{\alpha + T_t} - c_1 I_t T_t - d_1 I_t - a_1 I_t C_t \\
\dot{C}_t &= -d_2 C_t + u_t
\end{aligned}$$

The states are $\mathbf{x_t} = (N_t, T_t, I_t, C_t)$ where N_t is the number of normal cells, T_t is the number of tumor cells, I_t is the number of immune cells, and C_t is the drug concentration in the body. The action or control u_t [mg $\cdot\, l^{-1} \cdot$ day $^{-1}$] is the administered chemotherapy dose. To model the stochastic growth of tumor cells, the state updates are done thought the following SDE:

$$d\mathbf{x_t} = \mathbf{f}\left(\mathbf{x_t}, u_t\right) dt + \mathbf{G}\left(\mathbf{x_t} t\right) dW_t$$

Where $\mathbf{f}$ represents the deterministic dynamics given by the ODEs above, $\mathbf{G}$ is a constraint diffusion term only applied to T_t and W_t denotes a Weiner process commonly used to model cell motion.

The time horizon is $T = 60$ days, and updates happen at each day. The reward function is defined as $R_t = -dt \cdot (T_t + 0.5 \cdot [N_t < 0.4] + 0.5 \cdot [I_t < 0.4])$ where $[\cdot]$ is the Iverson bracket. It penalises having a high number of tumor cells as well as a low number of normal and immune cells. The states are initialized following $[1, 0.7, 1, 0]$. The constants have same value as in [7].

3 Methods

Using the OpenAI Gym library [1], we start by implementing the reinforcement learning environments corresponding to the mathematical modelisations in the previous section. We then used stable-baselines3 [8] that provides implementations of known Reinforcement Learning algorithms. Our implementation can be found at https://github.com/amine-abdeljaoued/RL_chemotherapy. Four algorithms were tried:

- DQN [6], Deep Q-Network: Q-Learning algorithm using several methods for stability: experience replay, target network, reward normalization and double Q-learning.
- PPO [9], Proximal Policy Iteration with clipped objective.
- DDPG [2], Deep Deterministic Policy Gradient: algorithm that does simultaneously Q-learning and policy learning.
- A2C [5], Advantage Actor-Critic: policy gradient method that works with simultaneous "agents" and averages the results periodically.

We trained the model until convergence or until a significant running time has elapsed for each algorithm. For DQN, as the algorithm requires discrete actions, we split the action space evenly in 10.

For our policy iteration and value function networks, we used a multi-layer perceptron (mlp) with ReLU as the activation function, containing 2 hidden layers of 64 units for the Patient-condition model and 128 units for the Cell-specific model. The increased parameter space is supposed to accommodate for the more complex model. For the Patient-condition model, the mean rewards for the 4 algorithms can be found in Figure 3. We also provide graphs with the mean length of episodes during training in Figures 4. For the Cell-specific model the mean rewards for the 4 algorithms can be found in Figure 5. The mean length of episodes has constant value of 60, which is the problem horizon.

4 Results

4.1 Patient-condition model [11]

From Figure 3, we can see that PPO clearly gives the best visual results. The rewards oscillates but they increase overall and tend to stay between -20 and -30 by the end. A2C performs poorly and both DQN and DDPG seem to learn a good policy but in a more noisy way. It is however worth noting that DDPG achieves much better mean rewards. In conclusion, PPO and DDPG seem to be the best options here. Let us now look at the obtained policies in Figure 1.

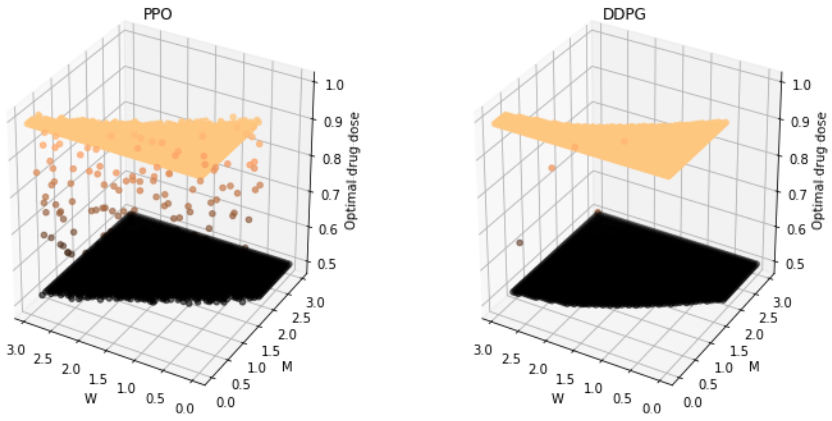


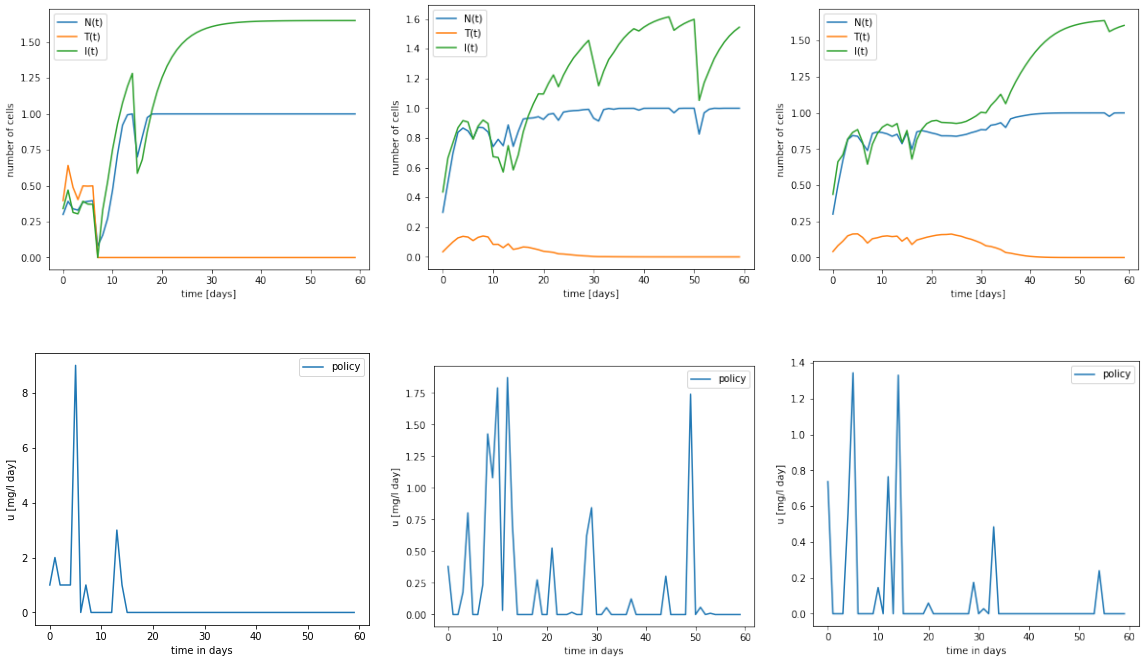
Figure 1: Resulting Policies

For small values of M and W, the optimal drug doses are the higher ones. The policy is then similar to a threshold one because the optimal drug doses only change when M and W increase to certain values (smooth change with values in between for PPO and a more abrupt one for DDPG). This is counter-intuitive because for small values of M, there should be no need to give a high dose as the tumor size is 0 or small. However here, with respect to W, the dosing is coherent as for small Ws (small toxicity), doses can be high given that the patient is well - but for high Ws it is the opposite. A possible reason here is that we forced our survival function to give more importance to wellness - but this should not pose such a big problem and this is also why we decided to explore chemotherapy dynamic models with more complicated and realistic transition functions.

4.2 Cell-specific model [7]

From Figure 5 all algorithms managed to converge to reasonable mean rewards. However when evaluating the policies, we realise that DDPG only predicts 0 (no drug administration). This is a commonly reported problem that has been investigated in [4]. A possible reason is that when the reward is found late (as it is the case in Fig. 5d), the process gets stuck with a sub-optimal policy. The authors in [4] claim that there is a critical window for finding the reward at the early stages of training.

Looking at the policies and simulated states in Figure 2 we notice a clear improvement going from DQN (fig. 2a) to A2C (fig. 2b) and finally to PPO (fig. 2c). The policy found by DQN eliminates the tumor cells faster by administrating a single large dose at the beginning, however this violates the thresholds set on N_t and I_t. In a real world scenario this could be a lethal dose. The PPO policy keeps the number of tumor cells under a threshold by administrating regular low doses. This also keeps N_t and I_t at high levels during the



(a) DQN Policy (b) A2C Policy (c) PPO Policy

Figure 2: Top: simulated states from cell dynamics in response to administered dose - N_t in green, I_t in blue, T_t in orange. [x-axis: time in days, y-axis: relative number of cells], Bottom: Resulting policies. [x-axis: time in days, y-axis: induces drug dose u_t

treatment. This is a much more realistic scenario. The A2C policy behaves similarly to PPO one, however it is much more erratic and sometimes administers unnecessarily high doses. Overall the PPO algorithm has found the best policy.

5 Conclusion

The two models capture the patient response to chemotherapy at two different scales and timeframes. The Patient-condition model majorly differs from the cell-specific model in that in incorporates patient's death, modelled by the survival function (which is in general cancer-specific). A Nonspecific model like we have used is less realistic thus making the results less interpretable. The obtained policy in our case is close to a threshold one (no dose or high dose). This issue is addressed in the cell-specific model, which provides a more precise approach to cancer dynamics and therefore results in a more realistic chemotherapy treatment policy. This study has also demonstrated the strengths and ease of use of the PPO algorithm.

References

[1] Greg Brockman et al. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[2] Timothy P. Lillicrap et al. "Continuous control with deep reinforcement learning". In: (2015). DOI: 10.48550/ARXIV.1509.02971. URL: https://arxiv.org/abs/1509.02971.

[3] Siqi Liu et al. "Reinforcement Learning for Clinical Decision Support in Critical Care: Comprehensive Review". In: *J Med Internet Res* 22.7 (July 2020), e18477. ISSN: 1438-8871. DOI: 10.2196/18477. URL: http://www.ncbi.nlm.nih.gov/pubmed/32706670.

[4] Guillaume Matheron, Nicolas Perrin, and Olivier Sigaud. "The problem with DDPG: understanding failures in deterministic environments with sparse rewards". In: *CoRR* abs/1911.11679 (2019). arXiv: 1911.11679. URL: http://arxiv.org/abs/1911.11679.

[5] Volodymyr Mnih et al. "Asynchronous Methods for Deep Reinforcement Learning". In: (2016). DOI: 10.48550/ARXIV.1602.01783. URL: https://arxiv.org/abs/1602.01783.

[6] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[7] Regina Padmanabhan, Nader Meskin, and Wassim M. Haddad. "Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment". In: *Mathematical Biosciences* 293 (2017), pp. 11–20. DOI: 10.1016/j.mbs.2017.08.004.

[8] Antonin Raffin et al. "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.

[9] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[10] *Understanding What is Cancer?* 2022. URL: https://www.cancer.gov/about-cancer/understanding/what-is-cancer.

[11] Yufan Zhao, Michael R. Kosorok, and Donglin Zeng. "Reinforcement learning design for cancer clinical trials". In: *Statistics in Medicine* 28.26 (2009), pp. 3294–3315. DOI: 10.1002/sim.3720.

Appendix

A Patient-condition reward function

The reward function is modelled by:

$$r_t = R\left(\mathbf{x}_t, a_t, \mathbf{x}_{t+1}\right) = R_1\left(D_t, W_{t+1}, M_{t+1}\right) + R_2\left(W_t, D_t, W_{t+1}\right) + R_3\left(M_t, D_t, M_{t+1}\right)$$

with:

- $R_1\left(D_t, W_{t+1}, M_{t+1}\right) = -60$, if patient died
- $R_2\left(W_t, D_t, W_{t+1}\right) = \begin{cases} 5 \text{ if } W_{t+1} - W_t \leq -0.5 \\ -5 \text{ if } W_{t+1} - W_t \geq -0.5 \\ 0 \text{ otherwise} \end{cases}$
- $R_3\left(M_t, D_t, M_{t+1}\right) = \begin{cases} 15 \text{ if } M_{t+1} = 0, \\ 5 \text{ if } M_{t+1} - M_t \leq -0.5, \text{ but } M_{t+1} \neq 0, \\ -5 \text{ if } M_{t+1} - M_t \geq 0.5, \\ 0 \text{ otherwise.} \end{cases}$

The death of a patient is modelled by a survival function $\Delta F(t)$ where the status (1 for death) is drawn from $Bernoulli(p)$ with $p = 1 - \Delta F(t)$.

We have:

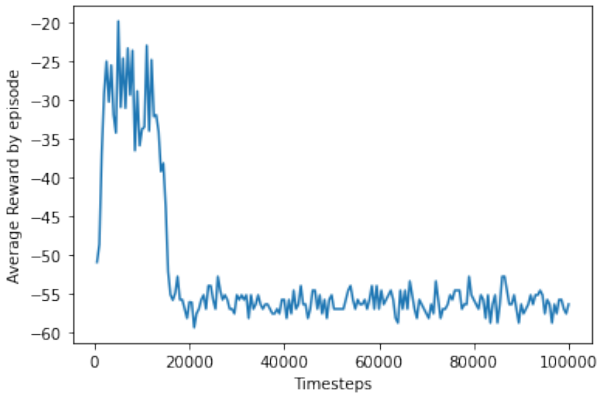
$$\Delta F(t) = \exp\left(-\Delta\Lambda(t)\right)$$

with:

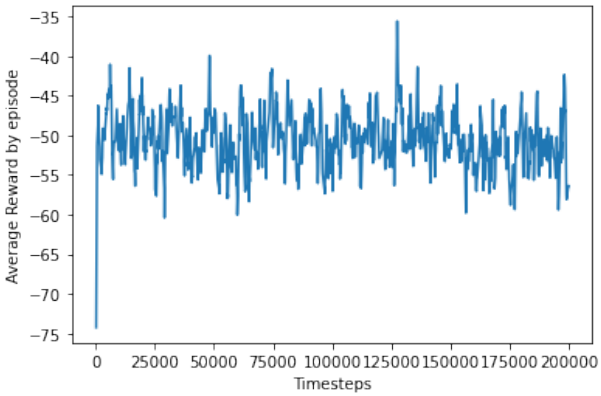
$$\Delta\Lambda(t) = \int_{t-1}^{t} \left[\exp(\mu_0 + \mu_1 W_s + \mu_2 M_s\right] d(s)$$

In our case $\Delta\Lambda(t)$ is computed using trapezoid integration. We chose the values $\mu = (\mu_0, \mu_1, \mu_2) = (-6, 2, 1)$ which give more importance to W_t.

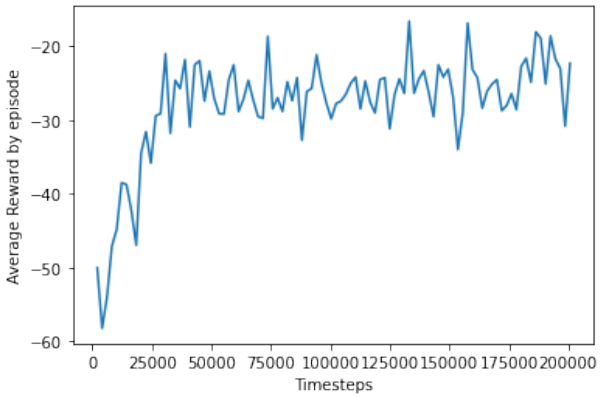
B Training rewards and lengths



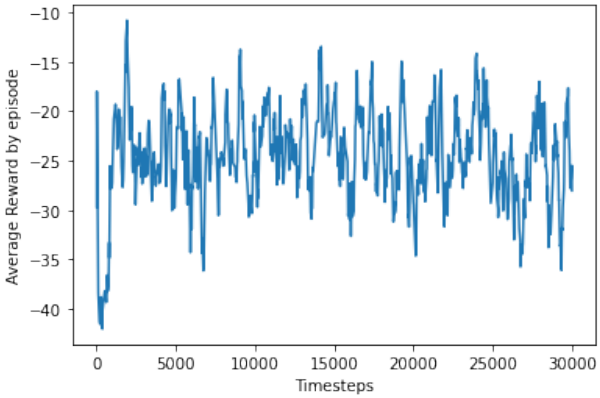
(a) A2C



(b) DQN

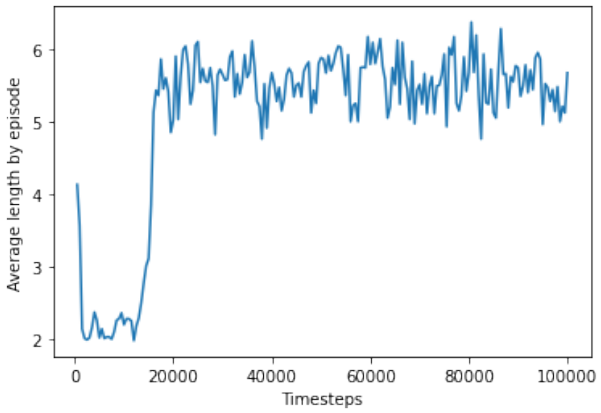


(c) PPO

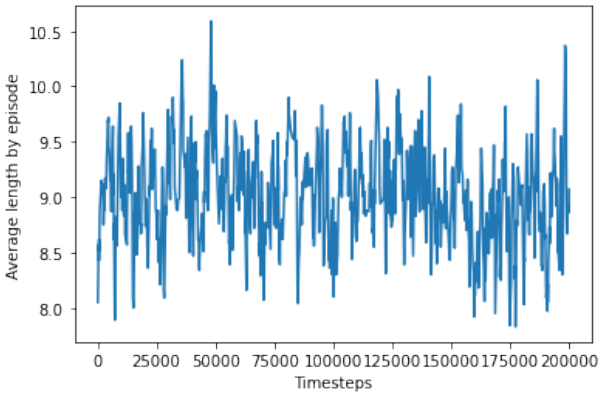


(d) DDPG

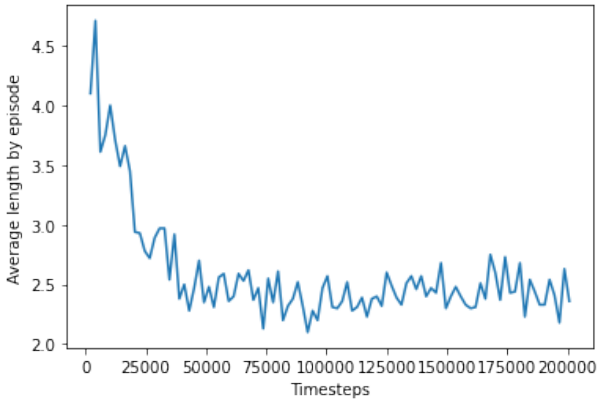
Figure 3: Patient-condition average rewards by episode through the timesteps for the different algorithms tried.



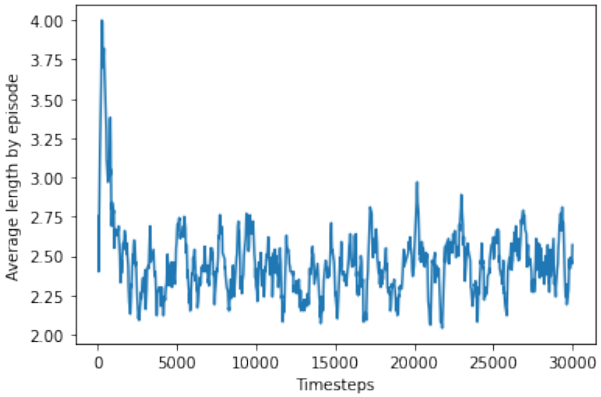
(a) A2C



(b) DQN

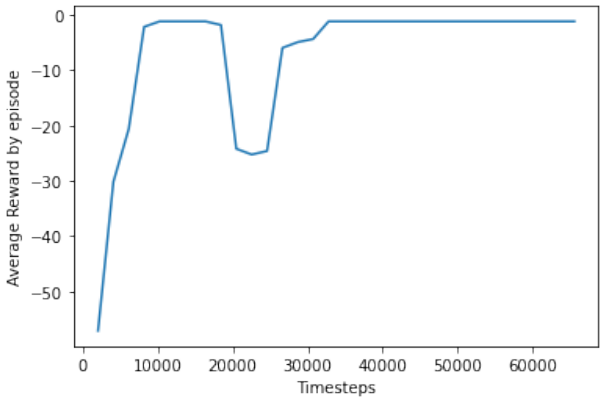


(c) PPO

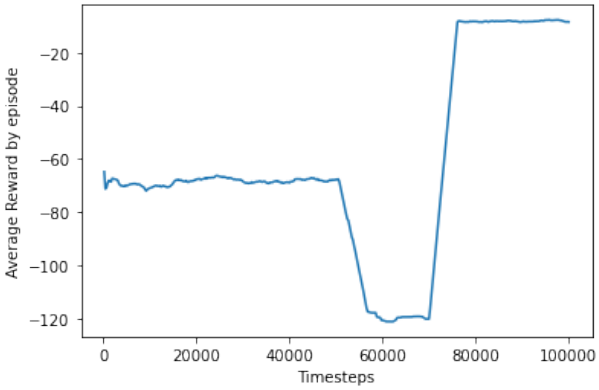


(d) DDPG

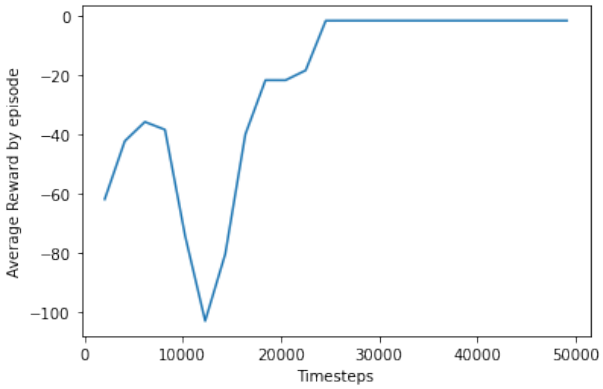
Figure 4: Patient-condition average length of episodes through the timesteps for the different algorithms tried.



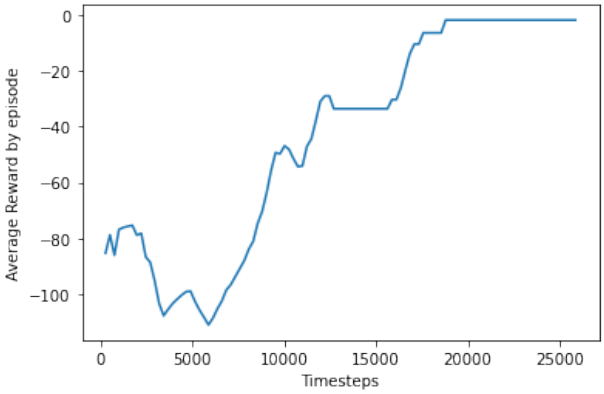
(a) A2C



(b) DQN



(c) PPO



(d) DDPG

Figure 5: Cell-specific average rewards by episode through the timesteps for the different algorithms tried.