Near On-Policy Experience Sampling in Multi-Objective Reinforcement Learning

Extended Abstract

Shang Wang
University of Washington
Tacoma, USA
swang848@outlook.com

Mathieu Reymond
Vrije Universiteit Brussel
Brussels, Belgium
mathieu.reymond@vub.be

Athirai A. Irissappane
University of Washington
Tacoma, USA
athirai@uw.edu

Diederik M. Roijers
Vrije Universiteit Brussel, Belgium & HU University of Applied Sciences Utrecht, the Netherlands
diederik.roijers@vub.be

ABSTRACT

In multi-objective decision problems, the same state-action pair under different preference weights between the objectives, constitutes different optimal policies. The introduction of changing preference weights interferes with the convergence of the network, and can even stop the network from converging. In this paper, we propose a novel experience sampling strategy for multi-objective RL problems, which samples transitions based on the weight and state similarities, to get the sampled experiences close to on-policy. We apply our sampling strategy in multi-objective deep RL algorithms on known benchmark problems, and show that this strongly improves performance.

KEYWORDS

Multiple Objectives; Reinforcement Learning; Experience Replay

ACM Reference Format:
Shang Wang, Mathieu Reymond, Athirai A. Irissappane, and Diederik M. Roijers. 2022. Near On-Policy Experience Sampling in Multi-Objective Reinforcement Learning: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION AND MOTIVATION

In *multi-objective reinforcement learning (MORL)*, multiple objectives are modeled explicitly as many – if not most – real-life problems may have multiple conflicting objectives [6, 19, 20]. For example, in operating a water reservoir one may aim to maximise hydro-power production, while minimising flood risk [4]. Such problems can be modeled as a *multi-objective Markov decision process (MOMDP)* [16, 21].

Often, we can model the importance of the objectives by assigning a weight to each them. Different weight vectors, $\mathbf{w}$, can then result in a different optimal policy [2]. Mossalam et al. [10] extend DQN [9] for the MOMDPs by learning vector-valued Q-functions for different weights. The *Conditioned Network (CN)* algorithm [1] improves the generalization ability of MORL across all weights, by training a single Q-Network that is conditioned on $\mathbf{w}$, and thus outputs weight-dependent Q-value vectors. Additionally, they introduce *Diverse Experience Replay (DER)*, a pruning strategy for the replay buffer that takes into account observed weights, as opposed to the first-in-first-out strategy used by the original replay buffer [9]. Abels et al. apply this algorithm in a *dynamic weights* setting [11], i.e., a setting in which the weights for the different objectives are provided by the environment and change over time. *Deep Conditioned Recurrent Actor-Critic (DCRAC)* [12] which extends CN – and uses DER – to an actor-critic approach to solve partially observable multi-objective problems (MOPOMDPs).

A key issue in the dynamic weights setting is the deteriorating performance when the weights change rapidly over time. This an issue for both CN [1] on fully observable MOMDPs and DCRAC [12] on MOPOMDPs. The inability to learn the optimal policies in such a regular weights-change setting is due to a problem known in off-policy reinforcement learning as the *extrapolation error* [5].

According to Fujimoto et al. [5], the extrapolation error can be attributed to a mismatch in the distribution of data induced by the policy and the distribution of data contained in the batch sampled from the experience replay buffer. In multi-objective dynamic weight settings, a buffer with transitions $(s, a, s', \mathbf{w}_{\text{old}})$ with older irrelevant weight-vectors $\mathbf{w}_{\text{old}}$, may not contain any state-action pair (s', a') that a policy derived from a value network conditioned on current weight-vector $\mathbf{w}_{\text{now}}$ would encounter. This results in large extrapolation errors. Similar observations were made in experiments [7] that compare old data from the experience replay buffer to the most recent data sampled under the behaviour policy.

To overcome the extrapolation error in the multi-objective dynamic weights setting, we introduce a novel sampling strategy called *Near-On Sampling Experience Replay (NER)* that assigns, for each transition in the experience replay buffer, a probability of being sampled from the buffer depending on its likelihood of being observed by the current policy. This is done by taking into account state-similarity and weight-similarity.

To evaluate the performance of our algorithm, we incorporate NER in CN and DCRAC, which are the state-of-the-art algorithms for MORL for MOMDPs, resp. MOPOMDPs. To the best of our knowledge, we are the first to design an experience sampling algorithm explicitly for MORL. Evaluation results on two well-known

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

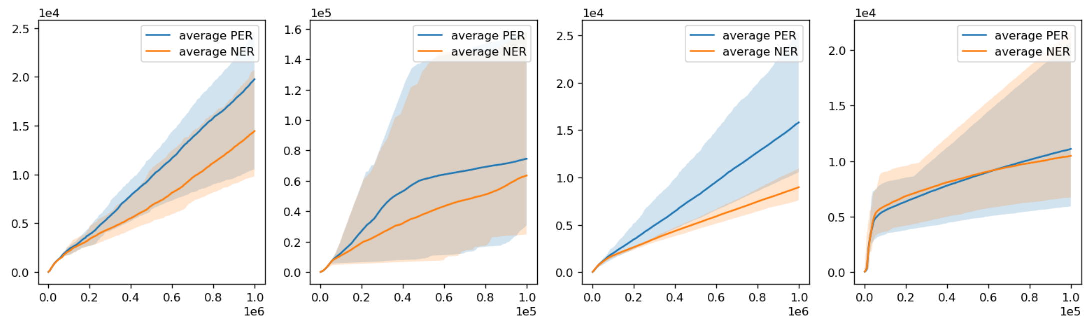

Figure 1: The average cumulative regret over 10 runs for the regular weights change in 4 different settings, from left to right, DCRAC in Minecart, DCRAC in DST, CN in Minecart, CN in DST.

MORL benchmarks, *Deep Sea Treasure* [18] and *Minecart* [1], show that our approach can efficiently reduce the extrapolation error in multi-objective reinforcement learning with dynamic weights.

2 ALGORITHM

To define similarity of the states with respect to the on-policy state distribution, we employ two metrics: the similarity between the current preference weight-vector and the preference weight-vector at the time the sampled transition was added to the replay buffer:

$$F(\mathbf{w}_j, \mathbf{w}_t) = ||\mathbf{w}_j - \mathbf{w}_t||_2. \quad (1)$$

as well as the similarity between the current states and the sampled transition's state. To compute state-similarity we employ a fixed CNN auto-encoder to embed the raw states into 512 dimensional randomized feature vectors $\psi(s)$, following [3]:

$$F(s_j, s_t) = ||\psi(s)_j, \psi(s)_t||_2. \quad (2)$$

Furthermore, to avoid the negative effects of sparse rewards, NER takes into account the TD-error, similarly to *Prioritized Experience Replay (PER)* [17]. The NER sampling algorithm is defined as follows: first, NER samples λk transitions from the replay buffer with the probability depending on the TD-error, as in [17]. Then, it computes a similarity score $F(\mathbf{w}_j, \mathbf{w}_t) + F(s_j, s_t)$ for each of these transitions, and makes a batch of the k most-similar transitions. Thus, if $\lambda = 1$, NER becomes equivalent to PER. Finally, the sampled transitions' priorities are updated by recomputing their TD-errors, as in [17].

Since the deviation from PER introduces a bias, we linearly anneal λ from an initial value λ_0 to 1 in $\alpha \cdot T$ steps, where T is the total number of training steps.

3 EXPERIMENTS AND EVALUATION

We evaluate the performance of NER on two well-known MORL benchmarks, *Minecart* [1] and *Deep Sea Treasure (DST)* [18]. Both CN [1] and DCRAC [12] originally used PER. We test what happens if we replace PER by NER. We employ the same basic network structure and hyper-parametersto those used in [1], [12].

We use the regret as evaluation metric, which is the difference between optimal value and actual return, $\Delta(\mathbf{g}, \mathbf{w}) = \mathbf{V}^*_{\mathbf{w}} \cdot \mathbf{w} - \mathbf{g} \cdot \mathbf{w} = \mathbf{V}^*_{w} \cdot \mathbf{w} - \sum_{t=0}^{T} \gamma^t \mathbf{r}_t \cdot \mathbf{w}$, where $\mathbf{g}$ is the discounted cumulative rewards,

avg regret / networks \ ER	PER	NER	PER	NER	PER	NER
	overall		last 250k steps		last 500 episodes	
Mine Cart						
CN	0.0158	**0.0089**	0.3488	**0.1511**	0.3756	**0.1821**
DCRAC	0.0198	**0.0144**	5.6318	**5.1626**	5.9456	**5.9257**
	overall		last 25k steps		last 50 episodes	
Deep Sea Treasure						
CN	0.1110	**0.1048**	0.4050	**0.3067**	0.5019	**0.2648**
DCRAC	0.7454	**0.6350**	6.5792	**5.8092**	8.5488	**5.7741**

Table 1: Average episodic regret with PER and NER for CN, DCRAC on both Minecart and DST.

and $\mathbf{V}^*_{\mathbf{w}}$ is the optimal value for $\mathbf{w}$. The weights $\mathbf{w}$ are randomly sampled form a Dirichlet distribution ($\alpha = 1$) every episode.

When looking at the performance (Table 1 and Figure 1) we see that CN/DCRAC+NER consistently outperforms CN/DCRAC+PER, both during training and after convergence. This strengthens our hypothesis that using data updates that conform to the current policy distribution is beneficial to improve performance.

4 CONCLUSION

In this paper, we studied the issue of steeply declining performance in the multi-objective reinforcement learning for dynamic weights when these weight-changes occur frequently. We argued that a current policy-network is more likely to adjust to new weights in a short time when the sampled transitions are from the state-distribution close to that of the current policy for the new weights. Therefore, we proposed a novel experience sampling strategy for multi-objective RL with dynamic weights, Near-On Sampling Experience Replay (NER). The results shows that our algorithm strongly outperforms standard prioritized experience replay (PER) [17] in the Minecart environment and the DST environment. NER alleviates the extrapolation error problem caused by the single-objective experience replay algorithms for MORL with dynamic weights. In future work, we aim to use NER in combination with different algorithms [8, 15], and in multi-objective multi-agent settings [13, 14].

ACKNOWLEDGMENTS

This research was supported by funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.

REFERENCES

[1] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. 2019. Dynamic weights in multi-objective deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 11–20.

[2] Leon Barrett and Srini Narayanan. 2008. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*. 41–47.

[3] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. 2018. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355* (2018).

[4] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. 2012. Tree-based fitted Q-iteration for multi-objective Markov decision problems. In *The 2012 international joint conference on neural networks (IJCNN)*. IEEE, 1–8.

[5] Scott Fujimoto, David Meger, and Doina Precup. 2018. Off-Policy Deep Reinforcement Learning without Exploration. *CoRR* abs/1812.02900 (2018). arXiv:1812.02900 http://arxiv.org/abs/1812.02900

[6] Conor F. Hayes, Roxana Radulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel de Oliveira Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2021. A Practical Guide to Multi-Objective Reinforcement Learning and Planning. *CoRR* abs/2103.09568 (2021). arXiv:2103.09568 https://arxiv.org/abs/2103.09568

[7] Riashat Islam, Komal K. Teru, Deepak Sharma, and Joelle Pineau. 2019. Off-Policy Policy Gradient Algorithms by Constraining the State Distribution Shift. arXiv:1911.06970 [cs.LG]

[8] Johan Källström and Fredrik Heintz. 2019. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS, Montreal, Canada, May 13-14, 2019*. 1–7.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[10] Hossam Mossalam, Yannis M Assael, Diederik M Roijers, and Shimon Whiteson. 2016. Multi-objective deep reinforcement learning. *arXiv preprint arXiv:1610.02707* (2016).

[11] Sriraam Natarajan and Prasad Tadepalli. 2005. Dynamic preferences in multi-criteria reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*. 601–608.

[12] Xiaodong Nian, Athirai A Irissappane, and Diederik Roijers. 2020. DCRAC: Deep conditioned recurrent actor-critic for multi-objective partially observable environments. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 931–938.

[13] David O'Callaghan and Patrick Mannion. 2021. Tunable Behaviours in Sequential Social Dilemmas using Multi-Objective Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 1610–1612.

[14] Roxana Rădulescu, Patrick Mannion, Diederik M Roijers, and Ann Nowé. 2020. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems* 34, 1 (2020), 1–52.

[15] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. 2022. Pareto Conditioned Networks. In *Proceedings of the 21th International Conference on Autonomous Agents and MultiAgent Systems*. To appear.

[16] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[17] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (2015).

[18] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. 2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning* 84, 1 (2011), 51–80.

[19] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. 2018. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology* 20, 1 (2018), 27–40.

[20] Peter Vamplew, Benjamin J Smith, Johan Kallstrom, Gabriel Ramos, Roxana Radulescu, Diederik M Roijers, Conor F Hayes, Fredrik Heintz, Patrick Mannion, Pieter J K Libin, Richard Dazeley, and Cameron Foale. 2021. Scalar reward is not enough: A response to Silver, Singh, Precup and Sutton (2021). *arXiv preprint arXiv:2112.15422* (2021).

[21] C Ch White, CC III White, and KW Kim. 1980. Solution procedures for vector criterion Markov decision processes. *Large Scale Systems* 1 (1980), 129–140.