Designing multi-objective multi-armed bandits algorithms: a study [1]

Mădălina M. Drugan [a] Ann Nowé [a]

[a] *Artificial Intelligence Lab,*
Vrije Universiteit Brussels,
Pleinlaan 2, B-1050 Brussels, Belgium

Many real-world problems are inherently multi-objective environments with conflicting objectives. Multi-armed bandits is a machine learning paradigm used to study and analyse resource allocation in stochastic and noisy environments. We consider the classical definition for the multi-armed bandits where only one arm is played at a time and each arm is associated with fix equal range stochastic reward vectors. When arm i is played at time steps t_1,t_2,..., the corresponding reward vectors $\mathbf{X}_{i,t_1}$, $\mathbf{X}_{i,t_2}, \ldots$ are independently and identically distributed according to an unknown law with unknown expectation vector. The independence holds between the arms.

We design a novel multi-armed bandit framework [3] that considers multi-objective (or multi-dimensional) rewards and that imports techniques from multi-objective optimization into the multi-armed bandits algorithms. We call this framework *multi-objective multi-armed bandits* (MO-MABs).

Multi-objective MABs lead to important differences to the standard MABs. There could be several arms considered to be the best according to their reward vectors. Let's consider two order relationships. *Scalarization functions*, like linear and Chebyshev functions, transform the reward vectors into scalar rewards. *Pareto partial order* allows to maximize the reward vectors directly in the multi-objective reward space. By means of an example, we compare these approaches on a non-convex distribution of the best arms. We highlight the limitations of the linear scalarization functions for optimizing non-convex shapes. Linear scalarization is currently a popular choice in designing multi-objective reinforcement learning algorithms, like the multi-objective MDPs from [4] but these algorithms have the same limitations as scalarized MO-MABs in exploring non-convex shapes. We consider a variety of scalarisation functions, and compare their performance to our Pareto MAB algorithm.

We propose three regrets metrics for multi-objective MAB algorithms. A straightforward regret for scalarized multi-objective MAB transforms the regret vector into a value using scalarization functions. This regret measure, however, does not give any information on the dynamics of the multi-objective MAB algorithm as a whole. Multi-objective MAB algorithms should pull *all* optimal arms frequently. Therefore, we also introduce an *unfairness* indicator to measure the lack of variance in pulling the optimal arms, and it is especially useful in pointing out the weakness of scalarized multi-objective MAB in discovering and choosing a variety of optimal arms. An adequate regret definition for the Pareto MAB algorithm measures the distance between the *set* of optimal reward vectors and a suboptimal reward vector.

Pareto UCB1 algorithm uses the Pareto dominance relationship to store and identify all the optimal arms in each iteration. As initialization step, each arm is played once. Each iteration, for each arm, we compute the sum of its mean reward vector and its associated confidence interval. A Pareto optimal reward set $\mathcal{A}'$ is calculated from these resulting vectors. Thus, for all the non-optimal arms $\ell \notin \mathcal{A}'$, there exists a Pareto

[1] The full paper has been published in *Proceedings of the International Joint Conference on Neural Networks* (IJCNN'13), 2013.

optimal arm $i \in \mathcal{A}'$ that dominates, or it is better, the arm ℓ, $\bar{\mathbf{x}}_\ell + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_\ell}} \not\succ \bar{\mathbf{x}}_i + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_i}}$. We select uniformly at random an optimal arm from $\mathcal{A}'$ and pull it. Thus, by design, this algorithm is fair in selecting Pareto optimal arms. After selection, the mean value $\bar{\mathbf{x}}_i$ and the common counters are updated. A possible stopping criteria is a maximum number of iterations.

The expected upper bound of the Pareto regret for Pareto UCB1 is logarithmic in the number of plays n, the number of dimensions D and the number of optimal arms $\mathcal{A}^*$. The worst-case performance of this algorithm is when the number of arms K equals the number of optimal arms $|\mathcal{A}^*|$. The algorithm reduces to the standard UCB1 for $D = 1$. Then, in most of the cases, $|\mathcal{A}^*| \approx 1$. In general, this Pareto UCB1 performs similarly with the standard UCB1 for small number of objectives and small Pareto optimal sets.

Two scalarization multi-objective variants of the UCB1 classical multi-armed bandits [1, 2] are proposed. These UCB1 algorithms assume a set of scalarizated reward vectors $S = (f^1, \ldots, f^S)$, $S \geq 1$, with different weights. The first algorithm is a straightforward generalization of the single-objective UCB1 that arbitrarily alternates different scalarization-based UCB1s. In initialization, each scalarization function from S and each arm is considered once. Until a stopping criteria is met, choose uniformly at random a scalarization function from S and run the corresponding scalarized UCB1. Let n^j be the number of times the function f^j is pulled, and let n_i^j be the number of times the arm i under function f^j is pulled. Let $\mathbb{E}[f^j(\bar{\mathbf{x}}_i)]$ be the expected reward of arm i under the scalarization function f^j. Given a scalarization function f^j, pull the arm that maximizes the term $\mathbb{E}[f^j(\bar{\mathbf{x}}_i)] + \sqrt{2\ln(n^j)/n_i^j}$, and update the counters. Update the counters and the expected value of $\bar{\mathbf{x}}_i = (\bar{x}_i^1, \ldots, \bar{x}_i^D)$. Note that each scalarized UCB1 has its own counter, such that the individual expected value for each arm is updated separately. Therefore, the upper scalarized regret bound is the same as in [1] and the upper bound for the scalarized regret of the scalarized multi-objective UCB1 is the sum of all upper bounds of the scalarized UCB1s.

In the case we can assume the Pareto front is convex and bounded we can use Lizotte et al [4]'s method, and obtain the minimum set of weights needed to generate the entire Pareto front. Then, the scalarized multi-objective UCB1 is fair in selecting the Pareto optimal arms. In a general setup, where the shape of the Pareto optimal sets is unknown, several sets of weights should be tried out in a scalarized multi-objective UCB1. Consider linear scalarization functions. Not all the reward vectors from *any* Pareto optimal reward set are reachable with this scalarization and, thus, there will be always a positive regret between the true and the identified Pareto optimal set of rewards. The unfairness of this algorithm is increasing with the number of plays because an arm identified as optimal is increasingly pulled whereas other optimal arms that are not recognized as optimal are scarcely pulled. A possible fix to these problems is given in the second scalarized UCB1 algorithm that is an improved UCB1 that removes scalarization functions considered not to be useful.

We compare runs of the proposed multi-objective UCB1 algorithms on multi-objective Bernoulli reward distributions, the standard stochastic environment used to test multi-armed bandits. Pareto UCB1 performs the best and is the most robust from the tested algorithms. To conclude, our Pareto UCB1 algorithm is the most suited to explore/exploit the multi-arm bandits with reward vectors.

References

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2/3):235–256, 2002.

[2] P. Auer and R. Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[3] M. M. Drugan and A. Nowé. Designing multi-objective multi-armed bandits algorithms: a study. In *Proc of Joint International Conference on Neural Networks*. IEEE, 2013.

[4] D.J. Lizotte, M. Bowling, and S.A. Murphy. Efficient reinforcement learning with multiple reward functions for randomized clinical trial analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.