Multi-objective Reinforcement Learning*

Manuela Ruiz-Montiel

June 26, 2013

Here we present PQ-learning, a new Reinforcement Learning (RL) algorithm that determines the rational behaviours of an agent in multi-objective domains. Most RL techniques focus on environments with scalar rewards. However, many real scenarios are best formulated in multi-objective terms: rewards are vectors and each component stands for an objective to maximize. In scalar RL, the environment is formalized as a Markov Decision Problem, defined by a set S of states, a set A of actions, a function $P_{sa}(s')$ (the transition probabilities) and a function $R_{sa}(s')$ (the obtained scalar rewards). The problem is to determine a *policy* $\pi : S \rightarrow A$ that maximizes the *discounted accumulated reward* $R_t = \Sigma_{k=0}^{\infty}\gamma^k r_{t+k+1}$. E.g., Q-learning [1] is an algorithm that learns such policy. It learns the scalar values $Q(s,a) : S \times A \rightarrow \mathbb{R}$, that represent the expected accumulated reward when following a given policy after taking a in s. The selected action a in each state is given by the expression $argmax_a Q(s,a)$. In the multi-objective case the rewards are vectors $\overrightarrow{r} \in \mathbb{R}^n$, so different accumulated rewards cannot be totally ordered; $\overrightarrow{v}$ dominates $\overrightarrow{w}$ when $\exists i : v_i > w_i \wedge \nexists j : v_j < w_j$. Given a set of vectors, those that are not dominated by any other vector are said to lie in the *Pareto front*. We seek the set of policies that yield non-dominated accumulated reward vectors.

The literature on multi-objective RL (MORL) is relatively scarce (see Vamplew et al. [2]). Most methods use preferences (lexicographic ordering or scalarization) allowing a total ordering of the value vectors, and approximate the front by running a scalar RL method several times with different preferences. When dealing with non-convex fronts, only a subset of the solutions is approximated. Some multi-objective dynamic programming (MODP) methods calculate all the policies at once, assuming a perfect knowledge of $P_{sa}(s')$ and $R_{sa}(s')$. We deal with the problem of efficiently approximating all the optimal policies at once, without sacrificing solutions nor assuming a perfect knowledge of the model. As far as we know, our algorithm is the

*This work is partially funded by: grant TIN2009-14179 (Spanish Government, Plan Nacional de I+D+i) and Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech. Manuela Ruiz-Montiel is funded by the Spanish Ministry of Education through the National F.P.U. Program.

first to bring these features together. As we aim to learn a set of policies at once, Q-learning is a promising starting point, since the policy used to interact with the environment is not the same that is learned. At each step, Q-learning shifts the previous estimated Q-value towards its new estimation: $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a'))$. In PQ-learning, Q-values are sets of vectors, so the max operator is replaced by $ND(\bigcup_{a'} Q(s',a')))$, where ND calculates the Pareto front. A naive approach to perform the involved set addition is a pairwise summation (imported from MODP methods), but it leads to an uncontrolled growth of the sets and the algorithm becomes impractical, as it sums vectors that correspond to different action sequences. The results of these mixed sums are useless when learning deterministic policies, because two sequences cannot be followed at once. We propose a controlled set addition that only sums those pairs of vectors that correspond to useful action sequences. This is done by associating each vector $\overrightarrow{q}$ with two data structures with information about the vectors that (1) have been updated by $\overrightarrow{q}$ and (2) have contributed to its value. PQ-learning has been applied to two problems of a benchmark [2]. It approximates all the policies in the true Pareto front (this is reflected by the hypervolume of the approximated front, Fig. 1), as opposed to the naive approach, that yields huge fronts with useless values that dramatically slow down the process.

References

[1] Watkins, C.J.: Learning from delayed rewards. PhD thesis, University of Cambridge (1989)

[2] Vamplew, P. et al.: Empirical evaluation methods for multiobjective reinforcement learning algorithms. Mach. Learn. **84**(1-2) (2011) 51–80

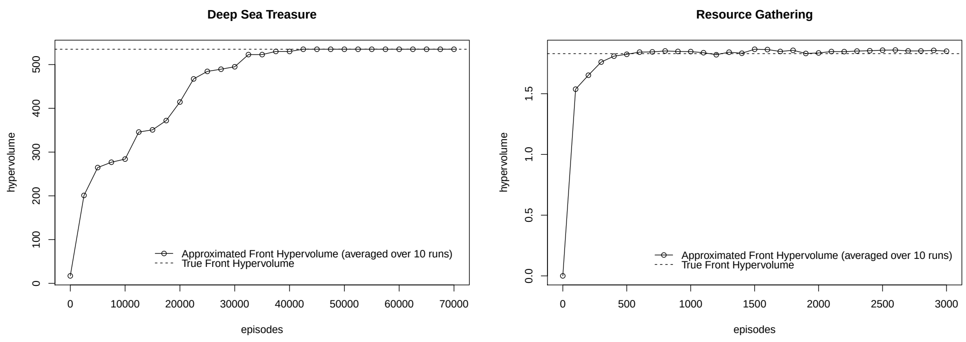

Figure 1: Results of PQ-learning over two MORL problems