Adaptive Objective Selection for Correlated Objectives in Multi-Objective Reinforcement Learning

(Extended Abstract)

Tim Brys
Vrije Universiteit Brussel
timbrys@vub.ac.be

Kristof Van Moffaert
Vrije Universiteit Brussel
kvmoffae@vub.ac.be

Ann Nowé
Vrije Universiteit Brussel
anowe@vub.ac.be

Matthew E. Taylor
Washington State University
taylorm@eecs.wsu.edu

ABSTRACT

In this paper we introduce a novel scale-invariant and parameterless technique, called *adaptive objective selection*, that allows a temporal-difference learning agent to exploit the correlation between objectives in a multi-objective problem. It identifies and follows in each state the objective whose estimates it is most confident about. We propose several variants of the approach and empirically demonstrate it on a toy problem.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Miscellaneous

General Terms

Algorithms, Performance

Keywords

Reinforcement Learning, Multi-Objective Optimization, Adaptive Objective Selection

1. INTRODUCTION

Most research on multi-objective optimization is focussed on solving problems with conflicting objectives, and rightly so, as these are hard problems with possibly many Pareto-optimal trade-off solutions. Problems with correlated objectives are usually dismissed as being easy [3], and therefore somewhat neglected. Yet, in [1], we demonstrate that in traffic light optimization, the objectives, delay and throughput, are correlated, and that a reinforcement learning agent can benefit from combining these signals, instead of using only a single one of these.

Generally, the problem class considered in this paper consists of those multi-objective reinforcement learning problems that have such strongly correlated objectives, and thus such a small Pareto front, that the system designer does not care about which of the very similar optimal trade-offs is learned, but rather how fast a (near-) optimal policy is found, and how close to optimality it is.

A solution technique often employed in multi-objective reinforcement learning is a linear scalarization of the objectives, reducing them to a single scalar objective using a weighted sum. Yet, these weights are hard to set *a priori* [2], and often require intensive weight tuning. In the problem class considered here, weight tuning is required to compensate for the potentially different scalings of the objectives.[1] Furthermore, weights are typically set globally, while we argue weights should be a function of state, as certain objectives may be more informative or reliable in some states than others. Therefore, we propose adaptive objective selection, a technique that addresses the issue of scaling, as well as making its decisions a function of state.

2. ADAPTIVE OBJECTIVE SELECTION

The basic idea of adaptive objective selection (AOS) is to estimate the Q-function of every objective o in parallel, and when action selection decisions need to be made, to determine for which objective the agent is most confident about the estimated Q-values. Only the estimates of this objective are then used to make an action selection decision. This automatically defines a dynamic, greedy weight function over the state space.[2]

We will propose several ways in which confidence in Q-values can be measured for a given state, but each of these comes down to representing each action as a distribution, and using a statistical test to check how significantly different the distributions of different actions are, or how much overlap there is between them. The better an agent can differentiate between the actions' distributions for a given objective, the more confident we say it is about that objective's estimates. The two key design decisions in AOS are then (1) how we represent the distribution of an action, and (2) how to test for difference between distributions.

To represent a distribution, one can either store a number of samples from that distribution, or store a parametric form of the distribution. In our case, we can keep track of the x

[1] As opposed to also specifying a desired trade-off between the objectives in general multi-objective problems.

[2] Note that this weighting function is implicit, and only applied at the action selection stage. It is greedy because only the estimates of a single objective are used.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

most recently observed $r(s,a,o) + max_{a'} Q(s',a',o)$ samples (sample-based representation). Or we can assume a normal distribution, using the Q-value as mean, and keep track of the variance of that distribution using δ_o, the TD-error for objective o (parametric representation):

$$VAR(s,a,o) = (1-\beta)VAR(s,a,o) + \beta\delta_o^2$$

Depending on the representation of the distribution, we can use a number of statistical tests to estimate confidence. Various statistical tests exist to indicate how significantly different distributions are based on a set of samples of each, such as the Student's t-test, the Wilcoxon signed rank test, ANOVA (analysis of variance), etc. The former two can only be applied to two distributions, e.g. the estimated best and worst actions' distributions. ANOVA can be applied to all actions' distributions at the same time. All of these tests calculate a p-value, which indicates how likely it is that the given estimates come from the same distribution. Confidence is inversely proportional to p.

If the distributions are represented in a parametric form, indicators such as the Bhattacharyya coefficient can be used to calculate the percentage of overlap between the distributions. Specifically, for normal distributions p and q, the Bhattacharyya coefficient $BC(p,q)$ can be calculated from the Bhattacharyya distance $BD(p,q)$):

$$BC(p,q) = \mathrm{e}^{-BD(p,q)}$$

$$BD(p,q) = \frac{1}{4}\ln\left(\frac{1}{4}\left(\frac{\sigma_p^4}{\sigma_q^4} + \frac{\sigma_q^4}{\sigma_p^4} + 2\right)\right) + \frac{1}{4}\left(\frac{(\mu_p - \mu_q)^2}{\sigma_p^4 + \sigma_q^4}\right)$$

Confidence is defined as $1 - BC(p,q)$, i.e the fraction of non-overlapping regions of the distributions.

Algorithm 1 shows the pseudocode for AOS. When intending to select an action, the agent first determines for which of the objectives it is most confident about its estimates, according to some confidence metric. Then, it makes an action selection decision (e.g. ϵ-greedy) based on that objective's estimates only. Given the scale-invariant nature of the statistical tests proposed, AOS itself is scale-invariant; its workings do not depend on any differences in scaling between the objectives. Also, the mechanism is completely automatic, as no parameters are introduced.

3. EXPERIMENTS

We evaluate AOS in a pathfinding problem situated in a 100×100 gridworld. The agent has two correlated objectives to optimize, each of which rewards a step towards the goal location and punishes a step away from the goal location. The objectives are differently scaled, and are subject to different levels of noise depending on the state. The RL agent is a Q-learning agent with ϵ-greedy action selection.

Figure 1 shows the results of a series of experiments on the pathfinding problem with increasing levels of noise and for various algorithm instances. We compare single-objective Q-learning, a linear scalarization with weights that align the objectives perfectly (simulating tuned weights), random objective selection, two AOS variants, and omniscient objective selection. The first AOS variant keeps a memory of 10 Q-samples and uses the Student's t-test to calculate confidence (MEM BW). The latter keeps track of the variance and uses the Bhattacharyya coefficient (VAR BW). Both use the estimated best and worst actions' distributions to measure confidence. Omniscient objective selection has perfect knowledge on what objective is least noisy in what state, and thus which one it can rely on most. Each data point plotted is the average of 100 runs, consisting of 300 episodes, each with a limit of 1000 steps. Both AOS techniques are able to learn faster than other tested techniques, without additional parameter tuning. They perform better than a linear scalarization with the best global weights, since they make their decisions dependent on state.

Algorithm 1 Adaptive Objective Selection

for each objective o **do**
  c_o = confidence$((s,a_1,o),...,(s,a_n,o))$
end for
$o_{best} = \arg\max_o c_o$
actionSelection$(Q(s,a_1,o_{best}),...,Q(s,a_n,o_{best}))$

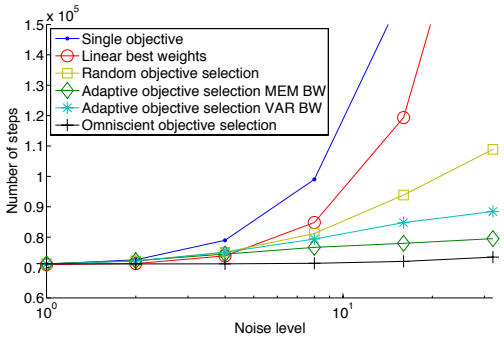

Figure 1: Average number of steps executed during a run of 300 learning episodes, shown for increasing differences in noise levels between objectives.

4. CONCLUSIONS

In this paper, we show that in problems with correlated objectives, learning can be improved by selecting actions based only on the estimates of the objective the agent is most confident about, without depending on the scaling of objectives or requiring parameter tuning.

Acknowledgement

Tim Brys is funded by a Ph.D grant of the Research Foundation Flanders (FWO). Kristof Van Moffaert is supported by the IWT-SBO project PERPETUAL (grant nr. 110041). This work was supported in part by NSF IIS-1149917.

5. REFERENCES

[1] T. Brys, T. T. Pham, and M. E. Taylor. Distributed learning and multi-objectivity in traffic light control. *Connection Science*, 2014.

[2] I. Das and J. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[3] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker. Response surface approximation of pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, 196(4):879–893, 2007.