Combining Multiple Correlated Reward and Shaping Signals by Measuring Confidence

Tim Brys and **Ann Nowé**
Vrije Universiteit Brussel
{timbrys, anowe}@vub.ac.be

Daniel Kudenko
University of York
daniel.kudenko@york.ac.uk

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

Abstract

Multi-objective problems with correlated objectives are a class of problems that deserve specific attention. In contrast to typical multi-objective problems, they do not require the identification of trade-offs between the objectives, as (near-) optimal solutions for any objective are (near-) optimal for every objective. Intelligently combining the feedback from these objectives, instead of only looking at a single one, can improve optimization. This class of problems is very relevant in reinforcement learning, as any single-objective reinforcement learning problem can be framed as such a multi-objective problem using multiple reward shaping functions. After discussing this problem class, we propose a solution technique for such reinforcement learning problems, called *adaptive objective selection*. This technique makes a temporal difference learner estimate the Q-function for each objective in parallel, and introduces a way of measuring confidence in these estimates. This confidence metric is then used to choose which objective's estimates to use for action selection. We show significant improvements in performance over other plausible techniques on two problem domains. Finally, we provide an intuitive analysis of the technique's decisions, yielding insights into the nature of the problems being solved.

Introduction

Multi-objective problems (MOP) require the simultaneous optimization of multiple feedback signals. As conflicts may exist between objectives, there is in general a need to identify (a set of) trade-off solutions. The set of optimal, i.e. non-dominated, incomparable solutions is called the Pareto-front. Assuming maximization, define the set $\mathcal{S}_p^*$ to contain all candidate solutions s of MOP p that are within ϵ_o of optimality for at least one objective o (w.r.t. utility function u_o):

$$s \in \mathcal{S}_p^* \iff \exists o \in \mathcal{O}_p, \forall s' \in p : u_o(s) + \epsilon_o \geq u_o(s')$$

$\epsilon_o \geq 0$ defines the largest difference in utility of objective o that the system designer is indifferent about. $\mathcal{S}_p^*$ will include at least the extrema of the Pareto-front of p.

We identify multi-objective problems with correlated objectives (CMOP) as a specific sub-class of multi-objective problems, defined to contain those MOPs p whose set $\mathcal{S}_p^*$ (and by extension whose Pareto-front) is so small that one can barely speak of trade-offs. By consequence, the system designer does not care about which of the very similar optimal solutions is found, but rather how fast it is found (and perhaps how well it is approximated). Formally:

$$p \in \text{ CMOP} \iff \forall o \in \mathcal{O}_p : \max_o(\mathcal{S}_p^*) - \min_o(\mathcal{S}_p^*) \leq \epsilon_o$$

Thus, whether a problem is a CMOP depends partially on the system designer's preferences (ϵ_o). Problems with every element in $\mathcal{S}_p^*$ achieving the same utility are contained in this class irrespective of the system designer's preferences (even $\forall o : \epsilon_o = 0$). Such problems can be seen as providing multiple sources of information or feedback for the same basic single-objective problem, and intelligently combining such objectives may yield faster and better optimization.

This paper deals with such reinforcement learning (RL) problems (Sutton and Barto 1998), formulated as *Correlated Multi-Objective Markov Decision Processes* (CMOMDP). (Single-objective) *MDPs* describe a system as a set of potential observations of that system's state S, a set of possible actions A, transition probabilities T for state-action-state triplets, and a reward function R that probabilistically maps these transitions to a scalar reward indicating the utility of that transition. The goal of an RL agent operating in an MDP is to maximize the expected, discounted return of the reward function. Temporal-difference learners such as SARSA (Rummery and Niranjan 1994) attempt this by estimating the Q-function, which represents the utility of each state-action pair. *MOMDPs* (Roijers et al. 2013) extend this framework to multiple objectives, with the reward function returning a vector of rewards to be maximized, and the added difficulty of finding trade-off solutions (e.g., trading off economical, environmental and aesthetic objectives in forest management (Bone and Dragićević 2009)). Finally, *CMOMDPs* satisfy the condition described above and remove the need for trade-offs (e.g., the traffic problem considered in this paper, where policies that minimize car delay simultaneously maximize the system's throughput).

The relevance of this problem class may seem limited, which can explain its neglect in the literature, but recent work on *multi-objectivization* opens up the possibility of framing *any* MDP as a CMOMDP. Multi-objectivization (Knowles, Watson, and Corne 2001) is the

Algorithm 1 Adaptive Objective Selection

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

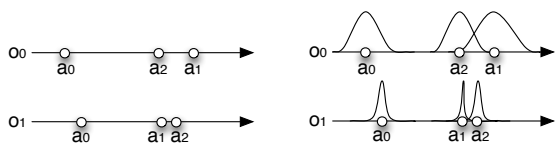
process of turning a single-objective problem into a multi-objective problem in order to improve solving of the single-objective problem. By preference, this new multi-objective problem is a CMOP, which would mean that no (significant) conflicts are introduced, and finding an optimal solution in the CMOP equals finding a (near-) optimal solution in the original single-objective problem. MDPs can simply be multi-objectivized by copying the reward function multiple times, which results in a CMOMDP with a single Pareto-optimal point. Of course, this modification in itself can not improve learning, but these copies of the basic reward signal can be diversified by adding a different potential-based reward shaping function to each (Brys et al. 2014). Since potential-based reward shaping is guaranteed to not alter the optimality of solutions (Ng, Harada, and Russell 1999), the problem remains a CMOMDP with a single Pareto optimal point, but each of the different shaping functions can help speed up learning differently. Their combined knowledge may be exploited by techniques specifically built for CMOMDPs — a linear scalarization or weighted sum is the simplest example (Devlin, Grześ, and Kudenko 2011).

This insight – that any MDP can be framed as a CMOMDP – significantly increases the importance of this problem class, as well as techniques developed for it, as these could potentially be used to solve regular MDPs *faster* and *better*, provided several meaningful shaping rewards can be devised. The remainder of this paper is devoted to proposing and evaluating such a technique, which introduces a way to measure confidence in learned estimates, and uses this confidence measure to combine the correlated signals.

Adaptive Objective Selection

Some authors working on multi-objectivization in evolutionary computation propose to make every optimization decision based on feedback from only a single of the objectives. Before every decision, they select one objective and use that to measure solution quality and accept/reject candidate solutions. This is possible since the objectives should strongly correlate with the original one. Jensen (2005) makes this objective selection decision uniformly at random, while Buzdalova and Buzdalov (2012) treat this selection as a dynamic multi-armed bandit problem, solving it with Q-learning.

We propose a similar approach for temporal-difference learners in CMOMDPs, called *adaptive objective selection*, see pseudocode in Algorithm 1. The learner estimates the Q-function for every objective o in parallel (thus learning $Q(s, a, o)$ values), and decides before every action selection decision which objective's estimates to use. To make this objective selection decision, we introduce the concept



$Q(s, a, o)$-values only (s, a, o) distributions

Figure 1: Showing estimates for two objectives (top and bottom respectively). Determining which objective's estimates an agent can be most confident about is impossible based on the estimated Q-values alone (left). Extra information is necessary (right). In this example, the second objective's estimates are deemed to be more reliable, as the actions' distributions are more significantly different/show less overlap.

of *confidence* in learned estimates. We define confidence as an estimation of the likelihood that the estimates are correct. Higher-variance reward distributions will make any estimate of the average reward less confident, and always selecting the objective whose estimates are most likely to be correct will maximize the likelihood of correctly ranking the action set. This is related to the use of confidence intervals in UCB (Auer, Cesa-Bianchi, and Fischer 2002). To measure confidence in estimates, the agent needs more information than simply the Q-values – how can one say whether the estimated Q-values for one objective are more likely to be correct than those for another objective, based only on those Q-values? If every (s, a, o)-triplet is modelled as a distribution on the other hand, and not just a mean (Q-value), then it becomes possible to determine how well each objective can differentiate between the actions based on common statistical tests. See Figure 1 for an illustration of this concept.

Key design decisions are then how to *represent* the (s, a, o)-triplets as meaningful distributions, and, depending on that representation, how to *measure* confidence.

Representation Distributions can be represented in a parametric form, or by keeping a number of samples from that distribution. An example of the former is to assume a normal distribution, with the value of $Q(s, a, o)$ as its mean, and to incrementally keep track of the variance of that distribution using the temporal-difference error (van Hasselt and Wiering 2007). An example of the latter is to store the n most recent $r(s, a, s', o) + \max_{a'} Q(s', a', o)$ samples (which represent the target being tracked).

Measurement Depending on the representation of the distribution, we can use a number of statistical tests to estimate confidence. Continuing with our examples, given mean and variance of two (assumed) normal distributions, we can calculate the Bhattacharyya coefficient (Bhattacharyya 1943), which indicates the percentage of overlap between given distributions. The less overlap between distributions, the better the agent can differentiate between the actions represented by those distributions. As this test can only be applied to two distributions, we suggest applying it to the estimated best and worst actions (according to the objective being evaluated), giving an indication of how well the agent can pull the action set apart.

For the latter example, differences for distributions represented by samples can be tested for statistical significance using such tests as the Student's t-test, the Wilcoxon signed-rank test, ANOVA, etc. These tests calculate a p-value which indicates how likely it is that the given estimates come from the same distribution. The smaller the p-value, the more likely it is the distributions are different, and that the agent can differentiate correctly between the actions.

Adaptive objective selection has several interesting properties. It makes its decisions a function of the state-space, which can account for different objectives being more or less reliable in different parts of the state space. Furthermore, it uses the objectives in a scale-invariant way. That is, its workings do not depend on the relative scalings of the objectives, since all statistical tests proposed are scale-invariant, and thus no parameters are introduced. This is a significant improvement over scalarization techniques (the most common approach to multi-objective problems), which usually require weight tuning, if only to align the magnitudes of the different correlated objectives in CMOPs. The optimality of the approach depends on the problem and the RL algorithm it is plugged into. For example, if a multi-objectivized problem is solved using Q-learning with adaptive objective selection, then the Q-learning guarantees make sure the estimates for every objective converge to the true values, and given the shaping guarantees, the greedy policies for each objective will be the same, and thus optimal.

The main disadvantage of the variants proposed above, is that extra memory is required to represent the (s, a, o) distributions, either for keeping track of variance, or for storing a set of samples for each triplet. In the following section, we introduce another variant that overcomes this disadvantage by exploiting the inherent Q-value decomposition of tile-coding function approximation to represent the distributions.

Adaptive Objective Selection using Tile-Coding Confidence

Many practical reinforcement learning problems have very large and/or continuous state spaces, making basic tabular learning methods impossible to use. A very popular way to overcome this problem is to use tile-coding function approximation (Albus 1981), which overlays the state space with multiple axis-parallel tilings. This allows for a discretization of the state-space, while the overlapping tilings guarantee a certain degree of generalization. The Q-function can then be approximated by learning weights that map the tiles activated by the current state s to an estimated Q-value:

$$\hat{Q}(s, a, o) = \theta_{o,a}^T \phi_s$$

ϕ_s is the feature vector representing state s, i.e. the tiles activated by this state, and θ is the parameter vector that needs to be learned to approximate the actual Q-function.

Recall that to apply adaptive objective selection, we need to represent (s, a, o) triplets as a distribution. Tile-coding provides a very natural way to do this, without requiring the storage of extra information. For a triplet (s, a, o), the agent can simply take the weights in $\theta_{o,a}$ activated by s as samples representing the distribution of that triplet. Then we can estimate confidence by applying a paired statistical test to the samples of every action (or of the estimated best and worst actions), see pseudocode in Algorithm 2. We can use a paired test, such as the paired Student's t-test or the Wilcoxon signed-rank test, because the weights for different actions and objectives will come from the same tiles, i.e. locations, in the same tilings, although stored in different weight vectors. This way, no extra memory is required to represent the distributions.

Algorithm 2 Adaptive Objective Selection using Tile-Coding Confidence

Require: State s
  for each objective o **do**
    //$\theta_{o,a}(\phi_s)$: the elements of $\theta_{o,a}$ activated by ϕ_s
    $p_o =$ paired test $(\theta_{o,a_1}(\phi_s), ..., \theta_{o,a_n}(\phi_s))$
  end for
  $o_{best} = \arg\min_o p_o$
  actionSelection$(\theta^T_{o_{best},a_1}\phi_s, ..., \theta^T_{o_{best},a_n}\phi_s)$

Test Domains

To demonstrate the potential of adaptive objective selection, we experimentally validate the variant that uses tile-coding and the paired Student's t-test to measure confidence on two CMOMDPs: the first represents problems that naturally fall into this problem class and the second is a multi-objectivized problem, illustrating how this technique can potentially be used to solve any MDP. We show results both for on-policy and off-policy learning.

Traffic Light Control

The problem of Traffic Light Control (Bazzan and Klügl 2013) consists of optimizing the timing schedules of a set of traffic lights in order to achieve (near-) optimal traffic flow. The two metrics most commonly used to measure performance are throughput and average delay. The former describes the number of cars passing through the system (to be maximized), the latter is the average delay experienced by the cars in the system (to be minimized). Given the strong correlation found between these two objectives in (Brys, Pham, and Taylor 2014), we classify this problem as a CMOMDP, and follow the same experimental setup. The experiments were implemented in the real-time AIM micro simulator (Dresner and Stone 2008), setup with a four intersection Manhattan grid. Each of the four lights is controlled by a separate SARSA(λ) agent, which has only local information, i.e. information about its own intersection. The agents act every two seconds, with two actions available: leaving the lights as they are, and changing the green direction (with a short period of yellow in between). The state space consists of three variables: 1) the time since the last 'change' action, 2) the time since the second to last 'change' action, and 3) a measure of the relative queue lengths in the green and red directions. Tile-coding is used to discretize the state space and all parameters are the same as in the original study, except $\epsilon = 0.05$, for better on-line performance.

Pursuit Domain

The Pursuit domain, or Predator/Prey, was proposed by Benda, Jagannathan, and Dodhiawala (1986) to investigate coordination mechanisms in a multi-agent system. The basic idea of pursuit is that a number of predators must capture a (number of) prey(s) by moving through a simple grid-world. Stone and Veloso (2000) identify many variants of the problem, and our implementation is as follows. There are two predators and one prey, and these can move in the four cardinal directions as well as choose to stay in place. The prey is caught when a predator moves onto the same gridworld cell as the prey; predators are not allowed to share the same cell. The prey takes a random action 20% of the time, with the rest of the time devoted to moving away from the predators. To do that, it takes the action that maximizes the summed distance from both predators, making the problem harder than with a fully random prey.[1] The predators are controlled by $Q(\lambda)$-learning agents, and both receive a reward of 1 when the prey is caught by one of them, and a reward of 0 the rest of the time. The predators observe the relative x and y coordinates of the other predator and the prey. Tile-coding is used to discretize the state-space, with 32 tilings, and tile-width 10, hashed down to 4096 weights. Action selection is ϵ-greedy, with $\epsilon = 0.1$. Further parameters are $\gamma = 0.9$, $\lambda = 0.9$ and $\alpha = \frac{1}{10 \times 32}$.

We formulate a CMOMDP by multi-objectivizing the problem using three potential-based shaping functions:[2]

Proximity encourages a predator to move closer to the prey. Its potential function is defined as $\Phi_P(s) = -d(pred, prey)$, with d the manhattan distance.

Angle encourages the predators to move to different sides of the prey, encircling it. It is defined to maximize the angle between them and the prey to π: $\Phi_A(s) = \arccos(\frac{x \cdot y}{|x||y|})$, with x and y vectors pointing from the prey to the two predators respectively.

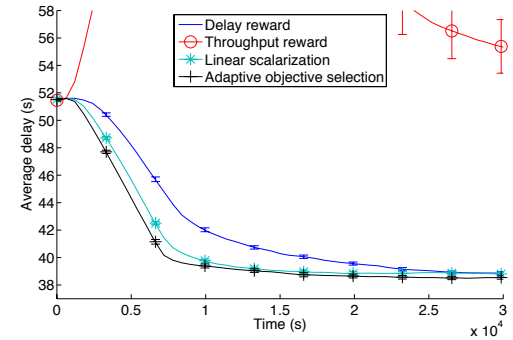
Separation encourages the predators to move away from each other. Its potential function is defined as $\Phi_S(s) = d(pred_1, pred_2)$ with d again the manhattan distance.

We will investigate both normalized and non-normalized shaping functions, as the magnitude of a shaping relative to the basic reward can have a significant impact on learning. Proximity and Separation are normalized by dividing by $2 \times size$, with $size = 20$ both the width and height of the world; Angle is normalized by dividing by π. Furthermore, Proximity is implemented as $2 \times size - d(pred, prey)$, so that all shaping functions are positive, and thus optimistic.[3]

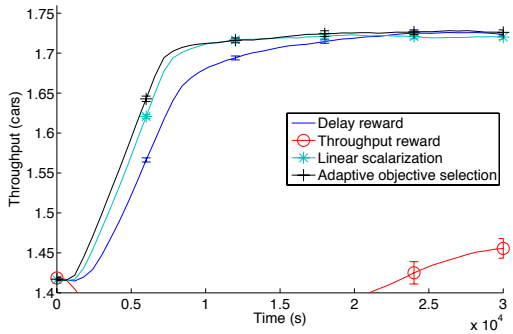
Results and Discussion

Traffic Light Control

The first CMOMDP we study is the traffic light control problem. We compare performance of single-objective learners learning on either delay or throughput as reward signal alone, on a linear scalarization of these two signals using tuned weights ($w_d = 0.04, w_t = 0.96$), and multi-objective learners employing adaptive objective selection. Figure 2 and Table 1 summarize the results of 100 runs of 3×10^4 in-simulation seconds each, with a 100 minute sliding average. Using the delay signal alone yields much better



(a) Average delay



(b) Throughput

Figure 2: Comparison of two single objective approaches with linear scalarization and adaptive objective selection. Errorbars indicate the 95% confidence interval (but are often too small to be seen). Adaptive objective selection learns faster and more consistently than the other variants.

Final reward	Delay	Throughput
Delay reward	38.91 ± 0.53	$\mathbf{1.72 \pm 0.062}$
Throughput reward	55.32 ± 10.08	1.46 ± 0.063
Linear scalarization	38.84 ± 0.69	1.72 ± 0.016
Ad. objective selection	$\mathbf{38.56 \pm 0.64}$	$\mathbf{1.73 \pm 0.016}$

Cumulative reward	Delay	Throughput
Delay reward	21747 ± 147	851 ± 2.86
Throughput reward	32617 ± 6483	703 ± 33.74
Linear scalarization	21154 ± 146	859 ± 3.09
Ad. objective selection	$\mathbf{20927 \pm 201}$	$\mathbf{862 \pm 4.58}$

Table 1: Final and cumulative performance achieved by the two single objective approaches, linear scalarization and adaptive objective selection. The best results and those not significantly different from the best (Student's t-test, $p > 0.05$) are indicated in bold.

[1] Results of our experimental validation are omitted for space.

[2] It has been proven that potential-based shaping in multi-agent RL does not alter the Nash Equilibria (Devlin and Kudenko 2011).

[3] The code used to run experiments in the pursuit domain can be downloaded at `http://ai.vub.ac.be/members/tim-brys`

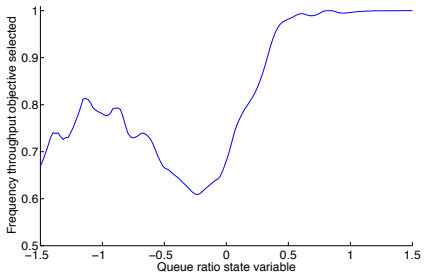


Figure 3: The frequency with which the throughput objective is selected, given specific queue-ratios during a single run of learning. $x = 0$ means equal queue lengths in the green and red directions, $x = 1$ means the queue length of the green direction is double that of the red, and $x = -1$ means the queue length of the red direction is double that of the green.

performance in this setup than using the throughput signal alone, which takes much longer to start improving. Adaptive objective selection automatically combines these two signals, outperforming the base signals and a scalarization with tuned weights in terms of final and cumulative performance, a measure of the speed of learning.

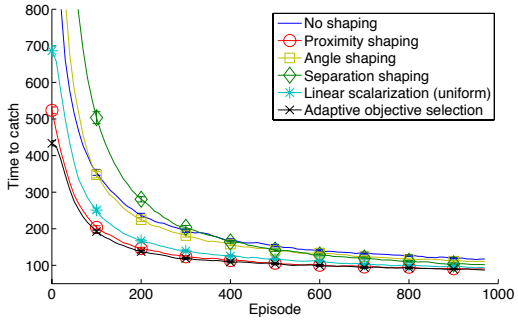
In Figure 3, we plot the fraction of times y throughput is selected as a function of the queue lengths in red and green directions, after learning has converged ($> 10^4 s$). Delay is selected the rest of the time ($1 - y$). We can infer from this plot how reliable each reward signal is estimated to be depending on the traffic distribution. Surprisingly, throughput most often yields the highest confidence, being selected over 70% of the time overall. Contrast this with the erratic learning behaviour displayed when using throughput alone; it performed the worst of all techniques. This graph suggests that throughput is the best predictor when the queue in the green direction is longer (selected up to 100% of the time), while estimates for delay are relatively more reliable when the queues are balanced, or unbalanced[4] with the queue in the red direction being longer, as these cars are then quickly accumulating delay.

Note that this single state variable (queue ratio) does not represent the whole state space, and that the other state variables also influence objective selection. However, this graph shows that, in practice, the relative reliability of delay and throughput can be correlated with the red-green queue ratio.

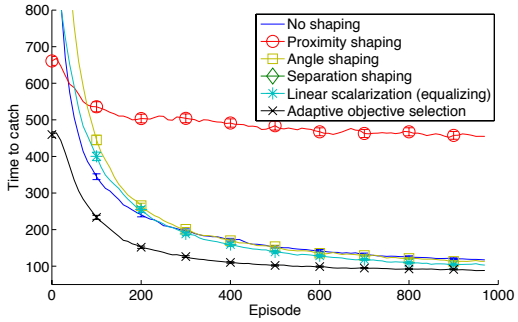
Pursuit Domain

This section demonstrates how adaptive objective selection can be used to solve a single-objective MDP, framed as a CMOMDP. We compare learning on the original reward, the original reward with the addition of a single shaping function, the original reward with a linear scalarization of the three shaping functions, and using adaptive objective selec-

[4]Unbalanced queues is not indicative of suboptimal policies. In that respect, balancing traffic is like pole balancing: it is impossible to keep it perfectly balanced all the time, but you can keep circling around the unstable equilibrium.



(a) Normalized shaping functions



(b) Non-normalized shaping functions

Figure 4: Comparison of no shaping, single shaping functions, a linear scalarization and adaptive objective selection, where the functions are either normalized or non-normalized (top and bottom respectively). Errorbars indicate the 95% confidence interval (but are often too small to be seen). Adaptive objective selection is always significantly better than the other variants, without requiring parameterization. It is also more robust with respect to the scaling (normalization) of the shaping functions themselves.

Final reward	Normalized	Non-normalized
No shaping	117 ± 2.67	117 ± 2.49
Proximity shaping	$\mathbf{90 \pm 2.11}$	452 ± 10.46
Angle shaping	110 ± 2.42	111 ± 2.37
Separation shaping	102 ± 2.15	1257 ± 44.95
Linear scalarization	94 ± 1.85	103 ± 2.11
Ad. objective selection	$\mathbf{88 \pm 1.94}$	$\mathbf{88 \pm 1.44}$

Cumulative reward	Normalized	Non-normalized
No shaping	211637 ± 2067	211581 ± 1985
Proximity shaping	131136 ± 1855	480046 ± 6493
Angle shaping	210428 ± 2117	242694 ± 3017
Separation shaping	251679 ± 3636	1368830 ± 31205
Linear scalarization	151763 ± 1189	202852 ± 2440
Ad. objective selection	$\mathbf{125668 \pm 1494}$	$\mathbf{134494 \pm 901}$

Table 2: Final and cumulative performance achieved by the single objective approaches, a linear scalarization of shaping functions, and adaptive objective selection. The best results and those not significantly different from the best (Student's t-test, $p > 0.05$) are indicated in bold.

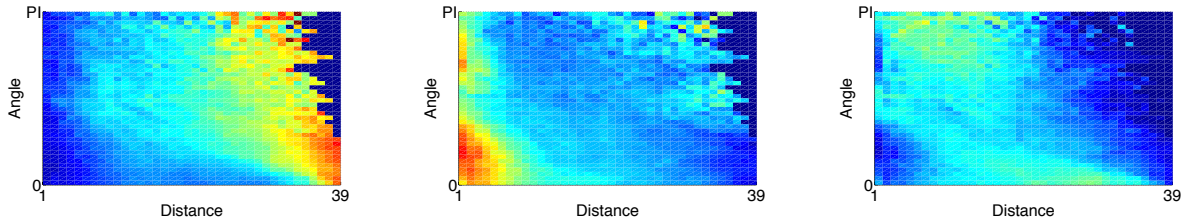


Figure 5: Selection of shaping functions as a function of the state space, with blue indicating 0% and red 100%. Results are averaged over 100 runs. Note that the blue region in the top right of the state space is such because it is simply never visited.

tion. We performed 1000 runs of 1000 episodes each, with a maximum number of steps per episode of 5000. Figure 4 and Table 2 summarize these results, using a sliding window of 25 episodes to smooth the graphs.

In the case of normalized shaping functions, using only the proximity shaping already gives great performance compared to the other variants, but adaptive objective selection is able to improve over these results still, learning even faster early on (statistically significant better cumulative performance). A linear scalarization with uniform weights yields performance in between the single shaping functions' performance, and tuning those weights yields no better performance than with the proximity shaping alone (results not included). Adaptively selecting which objective to use, or in this case which shaping, allows a learner to exploit the knowledge encoded in the shaping functions better than by simply combining them using global weights (i.e. the same scalarization weights for the whole state space). Global weights are in this case not enough to improve over single shaping functions, and weights should be defined as a function of state, which is even harder to do a priori than tuning a single set of global weights. Adaptive objective selection can be said to do this implicitly (and automatically), if we consider it to be a dynamic, greedy, state-dependent weight function applied at the action selection stage.

In the case of non-normalized shaping functions, the proximity and separation shaping functions are of too large a magnitude compared to the base reward and drown it, resulting in very bad performance. This indicates that these shaping functions do not completely correlate with the value function V^*. Using weights that equalize the magnitudes of the shaping functions[5] results in better performance, since the magnitude of the scalarized signal is not as large relative to the basic reward function. Despite the dramatic impact of the scaling of shaping functions on these other variants, adaptive objective selection barely suffers from this change, and has only slightly slower learning than with the normalized shaping functions. Because adaptive objective selection measures confidence, it can detect where a shaping function correlates less well with the basic reward, and select another for action selection. Note that we could not discover a better set of scalarization weights by tuning them.

In Figure 5, we show how often a predator selects each of the objectives (shaping functions) as a function of the distance to the prey, and the angle between the predators and the prey (0 meaning both predators are on the same side of the prey, π meaning predators are on opposite sides of the prey). The proximity shaping is selected most when a predator is far from the prey, the angle shaping (encouraging encircling) is selected most when the predator is close to the prey, and especially when it's on the same side of the prey as the other predator. The separation shaping is selected most at mid-range distances from the prey. This mapping between state and shaping makes a lot of intuitive sense, yet defining this manually a priori would be very hard to do. Adaptive objective selection on the other hand automatically discovered this mapping.

[5] $w_i = \frac{\prod_{j \neq i} s_j}{\sum_j \prod_{k \neq j} s_k}$, with s_i the size of the domain of shaping i.

Conclusions

We identified and formally defined a new class of multi-objective problems, called correlated multi-objective problems (CMOP), whose set of solutions optimal for at least one objective is so restricted that the decision maker does not care about which of these is found, but rather how fast one is found, or how well one is approximated. Such reinforcement learning problems (CMOMDPs) include traffic light control, with the delay and throughput objectives being strongly correlated, and, more significantly, any single-objective problem that has been multi-objectivized using multiple reward shaping functions.

After identifying this class of problems, we introduced a novel, parameterless and scale-invariant technique, called adaptive objective selection, that helps temporal-difference learners solve CMOMDPs faster and better. The variant we experimentally validated exploits the natural decomposition of Q-values by tile-coding to measure confidence in the estimates of each objective, using then only those estimates for an action selection decision. We validated the technique in traffic light control and in the pursuit domain, showing significantly improved learning. Additionally, the objective selection decisions yield intuitive insights into the problems.

We believe many more techniques exist (*in potentia*) to solve CMOMDPs. E.g., ensemble techniques for RL (Wiering and van Hasselt 2008) combine different learners for the same signal in order to create a learner that is better than any of its constituting parts. "The same signal" can be relaxed to the condition described in this paper for CMOPs, allowing their application to this type of problem.

Acknowledgments

Tim Brys is funded by a Ph.D grant of the Research Foundation-Flanders (FWO). This work was supported in part by NSF IIS-1149917 and NSF IIS-1319412.

References

Albus, J. 1981. *Brains, behavior and robotics*. McGraw-Hill, Inc.

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

Bazzan, A. L., and Klügl, F. 2013. Introduction to intelligent systems in traffic and transportation. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 7(3):1–137.

Benda, M.; Jagannathan, V.; and Dodhiawala, R. 1986. On optimal cooperation of knowledge sources - an empirical investigation. Technical Report BCS–G2010–28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, WA, USA.

Bhattacharyya, A. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc* 35(99-109):4.

Bone, C., and Dragićević, S. 2009. Gis and intelligent agents for multiobjective natural resource allocation: a reinforcement learning approach. *Transactions in GIS* 13(3):253–272.

Brys, T.; Harutyunyan, A.; Vrancx, P.; Taylor, M. E.; Kudenko, D.; and Nowé, A. 2014. Multi-objectivization of reinforcement learning problems by reward shaping. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE.

Brys, T.; Pham, T. T.; and Taylor, M. E. 2014. Distributed learning and multi-objectivity in traffic light control. *Connection Science* 26(1):56–83.

Buzdalova, A., and Buzdalov, M. 2012. Increasing efficiency of evolutionary algorithms by choosing between auxiliary fitness functions with reinforcement learning. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1, 150–155. IEEE.

Devlin, S., and Kudenko, D. 2011. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 225–232.

Devlin, S.; Grześ, M.; and Kudenko, D. 2011. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems* 14(02):251–278.

Dresner, K., and Stone, P. 2008. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research* 31:591–656.

Jensen, M. T. 2005. Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal of Mathematical Modelling and Algorithms* 3(4):323–347.

Knowles, J. D.; Watson, R. A.; and Corne, D. W. 2001. Reducing local optima in single-objective problems by multi-objectivization. In *Evolutionary Multi-Criterion Optimization*, 269–283. Springer.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, volume 99, 278–287.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48:67–113.

Rummery, G. A., and Niranjan, M. 1994. *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering.

Stone, P., and Veloso, M. 2000. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots* 8(3):345–383.

Sutton, R., and Barto, A. 1998. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press.

van Hasselt, H., and Wiering, M. A. 2007. Reinforcement learning in continuous action spaces. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007.*, 272–279. IEEE.

Wiering, M. A., and van Hasselt, H. 2008. Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 38(4):930–936.