A Flexible Approach for Designing Optimal Reward Functions

(Extended Abstract)

Ricardo Grunitzki
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
rgrunitzki@inf.ufrgs.br

Bruno C. da Silva
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
bsilva@inf.ufrgs.br

Ana L. C. Bazzan
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

ABSTRACT

Defining a reward function that, when optimized, results in the rapid acquisition of an optimal policy, is one of the most challenging tasks involved in applying reinforcement learning algorithms. The behavior learned by agents is directly related to the reward function they are using. Existing works on Optimal Reward Problem (ORP) propose mechanisms to design reward functions that facilitate fast learning, but their application is limited to specific sub-classes of single or multi-agent reinforcement learning problems. Moreover, while these methods identify which rewards should be given in which situation, they do not give clues regarding on which features of the state or environment should be used when defining a reward function. In this paper, we address these and other issues of ORP. Experimental results on a gridworld scenario are used to evaluate the efficacy of our approach in designing effective reward functions.

Keywords

Optimal reward problem, multi-agent reinforcement learning, multi-agent coordination

1. INTRODUCTION

In the traditional RL framework, a reward function reflects the goals of both the agent and the designer. Singh et al. [6, 7] suggest that an RL system can benefit by decomposing theses goals into two functions: a reward function that guides the learning process and a fitness function that evaluates the quality of the learned behavior. The ORP consists in finding a reward function that maximizes the fitness. Formally, at each time step, an agent G receives an observation $o \in \mathcal{O}$ from an environment M, takes action $a \in A$, produces a history h, composed of observations of states, actions and rewards, and repeats this process for a certain time horizon. The agent's goals are represented by a reward function R, which it tries to be maximize, while the designer's goals are represented by a fitness function F, which produces a cumulative return $F_R(h)$ for a reward function R over a history h. The Optimal Reward Function (ORF) problem is defined in Eq. 1, where R^* is an ORF that maximizes the designer's expected fitness F of agent G in M, and where the maximization is over a pre-defined space $\mathcal{R}$ of reward functions. The notation $h \sim M\langle G \rangle$ denotes that h is a sample history generated when agent G acts in environment M, and $G(R)$ denotes an agent G tasked with maximizing a reward function R.

$$R^* = \arg\max_{R \in \mathcal{R}} \mathbb{E}\left[F_{R_0}(h) \mid h \sim M\langle G(R) \rangle\right] \quad (1)$$

In [6, 8], the ORP is (approximately) solved through a brute force strategy. Subsequent works [8, 5, 4, 3] propose automated and more efficient methods for dealing with ORP in single or multi-agent settings. Such methods present the following limitations, which are addressed by our approach, called *Extended* Optimal Reward Problem (EORP): i) the situations (or state features) in which the agent may be rewarded must be pre-defined by the designer; ii) lack of generality for dealing with RL problems that may be either single and multi-agent settings; iii) scalability in multi-agent settings; iv) the trade-off between fitness and *learning effort* is not considered in the standard ORP.

2. OUR APPROACH

The EORP is defined in Eq. 2, where H is the set of all histories generated by agents $i \in I$ learning with a reward function R. Note that Eq. 2, unlike the standard ORP formulation, allows for a reward function to be designed that maximizes the expected fitness of multiple (possibly cooperative) agents, situated within a multi-agent system. The two main changes of this formulation with respect to the one in Eq. 1 are the introduction of: i) a reward design space, $\mathcal{R}(J)$, corresponding to the space of reward functions spanned by a given set of feature states J; and ii) a multi-objective function $\mathcal{F}$, called evaluation function.

$$R^* = \arg\max_{R \in \mathcal{R}(J)} \mathbb{E}\left[\mathcal{F}_{R_0}(H) \mid H \sim M\langle I(R) \rangle\right] \quad (2)$$

In the space of reward functions $\mathcal{R}(J)$, more than one function $R \in \mathcal{R}(J)$ can produce the same fitness F_R, but under different learning effort. For instance, one reward function may allow the agent to learn an optimal policy in fewer steps because it provides the agent with more informative guidelines about the effectiveness of its current behavior. For the designer, it is interesting to identify a reward function that, when optimized, results in the agent more rapidly learning to solve the task. The traditional ORP does not assist the designer in such decisions because it is only aimed

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

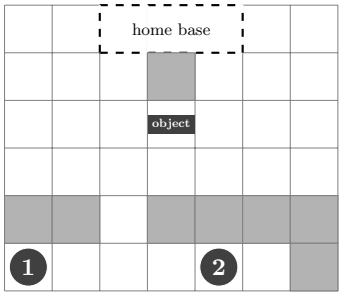

Figure 1: CMOTP scenario.

Table 1: Fitness (f_1) and effort (f_2) obtained by R^* and R^B on CMOTP.

	R^B	R^*
f_1	17.83 ± 1.74	12.4 ±0.72
f_2	37.97 ± 3.03	26.83 ± 1.29
j_0	1	0.77
j_1	0.1	0.86
j_2	-	-0.47
j_3	-	-
j_4	-	-0.08
j_5	-	-0.28
j_6	-	-
j_7	-	-
j_8	-	-0.59
j_9	0	-0.23

$$J = \begin{cases} j_0, & \text{if the object is at the home base;} \\ j_1, & \text{if the agent grasped the object;} \\ j_2, & \text{if the agent hit a wall;} \\ j_3, & \text{if the agent tried to move the object by itself;} \\ j_4, & \text{if the agent tried to go to a cell occupied by an agent;} \\ j_5, & \text{if the agent tried to go to a cell occuped by an object;} \\ j_6, & \text{if other agent tried to go to a same cell;} \\ j_7, & \text{if the agent executed uncoordinated action to move the object;} \\ j_8, & \text{if the agent chose to stand still;} \\ j_9, & \text{otherwise.} \end{cases} \quad (3)$$

at maximizing fitness. In the EORP, we consider that the designer has multiple goals to be maximized, such as fitness and learning effort. The function $\mathcal{F}$ evaluates the fitness (f_1) and effort (f_2) produced by H and returns a 2-dimensional vector $\mathcal{F}(H) = [f_1(H), f_2(H)]$. The fitness function measures the quality of the final learned behavior achieved when maximizing a given reward function, while the learning effort function evaluates the amount of effort spent by the agent (e.g., time until convergence) during its lifetime. The designer must define both functions to represent his goals for a given learning task.

The reward design space $\mathcal{R}(J)$ represents the set of all possible reward functions spanned by a given set of features J. A feature can be seen as a situation in which the agent may be rewarded. A reward function $R \in \mathcal{R}(J)$ is represented as $R = \sum_{k=1}^{J} s(k)\, w(k)\, P(k)$, where for each feature k, $s(k) \in \{0, 1\}$ indicates that the feature k is activated in state s or not, $w(k) = \{x \in \mathbb{R} \mid -1 \leq x \leq 1\}$ represents the contribution to the reward signal of k, and $P(k) = \{x \in \mathbb{N} \mid 0 \leq x \leq 1\}$ is an indicator function reflecting if k is used or not to compose R. Given a set of n features, a search space $\mathcal{R}(J) \subset \mathcal{R}^{2n} = [\{w(k_0), \ldots, w(k_n)\}, \{p(k_0), \ldots, p(k_n)\}]$ contains n indicator features $P(k)$ and n reward signals, $w(k)$. The number of decision variables of an EORP with such search space is $2n$. Our method optimizes a single reward function that is used by all collective of learning agents I interacting in a given problem. This way, independently of the amount of agents, the dimensionality of the optimization problem is always given by the amount of features ($|J|$) in the state space of one individual agent.

Solving the EORP consists in finding the set of Pareto optimal solutions, $R^* \in \mathcal{R}(J)$, that maximize $\mathcal{F}$. Any multi-objective optimization algorithm that deals with real and integer decision variables can be used. In this paper, we opted to use a multi-objective version of the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)[2]. EORF criterion does not depend on knowledge of the dynamics of the problem M, and so even model-free optimization algorithms can be used to optimize Eq. 2. This is unlike the approach by Liu et al. [4, 3], which requires model-based algorithms. Additionally, our method produces only one EORP solution per collective of agents, so that every agent $i \in I$ will learn with an instance of this function. In [4, 3], one reward function is optimized for each agent, which may present scalability issues in problems with a large amount of learning agents.

3. SCENARIO AND PARTIAL RESULTS

We evaluate our approach on the Coordinated Multi-agent Object Transportation Problem (CMOTP), presented in [1]. The learning task involves the coordinated transportation of an object by two agents represented in Fig. 1. The two agents (numbered circles) can move in a two-dimensional 7×6 grid, in order to transport an object to the home base in the least amount of time. The task involves the avoidance of obstacles (gray blocks) and coordinated moves. At each time step, they can move by one cell to the left, right, up, down or stand still at the current cell. The task is accomplished when the object reaches the home base.

In this setting, each agent maximizes its reward function using Q-Learning. The Q-learning agents, the MDP modeling and the parameters values were defined according to the specifications presented by Buşoniu et al.[1]. For comparison purposes, we used the reward function R^B proposed by the authors as a baseline. As stopping criteria for CMA-ES, we used 1000 evaluations of the fitness function.

Given the set of potential features in Eq. 3, the experiment consists in finding which features and reward signals produce an optimal reward function, R^*, that solves the task with maximum fitness and minimum effort. In this experiment, the fitness function evaluates the number of steps to accomplish the task at the current episode, and the learning effort function represents the cumulative number of decisions taken by all agents along their lifetimes.

In Tab. 1, we present the results obtained by solving the EORP and by applying R^B. In 30 repetitions, our method generated 102 Pareto optimal ORFs from which we selected R^* for comparison purposes. The reward function R^* performed better than the baseline regarding both fitness and effort. The fitness produced by R^* is closer to the optimal solution for the CMOTP, which is 12 steps. A reduction of $\approx 30\%$ in the learning effort is also observed when compared to R^B. Seven features were used in R^*, all of which automatically identified by the maximizing of Eq. 2. The features directly related to the success of the task, j_0 and j_1, are the only ones to receive positive reward signals. The rest of the features utilized in R^* are associated with negative reward signals that discourage those situations. This experiment shows that the EORP can identify the features and reward signals that compose an optimal reward function which, when optimized, produces maximum fitness and minimum effort.

4. NEXT STEPS

As future work, we intend to perform an evaluation in more complex scenarios, with more agents and potential reward features. Investigating theoretical properties of our method, resulting from using methods that find Pareto optimal solution, is also part of our future work.

REFERENCES

[1] L. Buşoniu, R. Babuška, and B. De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in Multi-Agent Systems and Applications-1*, pages 183–221. Springer, 2010.

[2] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28, 2007.

[3] B. Liu, S. Baveja, R. Lewis, and S. Qin. Optimal Rewards for Cooperative Agents. *Autonomous Mental Development, IEEE Transactions on*, 11(4), 2014.

[4] B. Liu, S. Singh, R. L. Lewis, and S. Qin. Optimal rewards in multiagent teams. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics, ICDL 2012*, 2012.

[5] S. Niekum, A. Barto, and L. Spector. Genetic Programming for Reward Function Search. *IEEE Transactions on Autonomous Mental Development*, 2(2):83–90, 2010.

[6] S. Singh, R. L. Lewis, and A. G. Barto. Where do rewards come from. In *Proceedings of the annual conference of the cognitive science society*, pages 2601–2606, 2009.

[7] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *Autonomous Mental Development, IEEE Transactions on*, 2(2):70–82, 2010.

[8] J. Sorg, R. L. Lewis, and S. P. Singh. Reward Design via Online Gradient Ascent. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2190–2198. Curran Associates, Inc., 2010.