Niekum, S., L. Spector, and A. Barto. 2011. Evolution of Reward Functions for Reinforcement Learning. In *GECCO'11 Posters, Genetic and Evolutionary Computation Conference*. ACM Press. pp. 177-178.

Evolution of Reward Functions for Reinforcement Learning

Scott Niekum
Computer Science
University of Massachusetts
Amherst, MA 01003 USA
sniekum@gmail.com

Lee Spector
Cognitive Science
Hampshire College
Amherst, MA 01002 USA
lspector@hampshire.edu

Andrew Barto
Computer Science
University of Massachusetts
Amherst, MA 01003 USA
barto@cs.umass.edu

ABSTRACT

The reward functions that drive reinforcement learning systems are generally derived directly from the descriptions of the problems that the systems are being used to solve. In some problem domains, however, alternative reward functions may allow systems to learn more quickly or more effectively. Here we describe work on the use of genetic programming to find novel reward functions that improve learning system performance. We briefly present the core concepts of our approach, our motivations in developing it, and reasons to believe that the approach has promise for the production of highly successful adaptive technologies. Experimental results are presented and analyzed in our full report [3].

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming—*Program synthesis*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Reinforcement learning, genetic programming, Push, PushGP, hungry-thirsty problem

1. PRIOR WORK

Reinforcement learning is a machine learning paradigm that addresses problems in which an agent seeks to maximize cumulative rewards in environments that provide no teacher or supervisor and in which rewards may be significantly delayed [9]. Reinforcement learning systems have achieved notable successes in a wide range of application areas including robotics, control, operations research, games, and human-computer interaction. They are driven by immediate environmental reward signals that are usually derived directly from the descriptions of the problems that the systems are being used to solve. However, the directness of this derivation is not mandated by reinforcement learning theory and indeed the algorithms for reinforcement learning can in some cases learn faster or more effectively when using different reward functions. For example, Ng, Harada, and Russell [2], described a class of 'shaping' reward adjustments and showed that such adjustments could improve learning system performance. But they did not specify a mechanism for deriving such beneficial adjustments for particular domains or problems, and practitioners have found that it can be difficult to design good reward functions "by hand."

We have explored the opportunities for the use of evolutionary computation in the search for reward functions that improve the performance of reinforcement learning systems. We have done so in the context of recent work by Singh, Lewis, and Barto [5, 6] demonstrating that for some problems there exist a wide spectrum of reward functions with different properties, and that the apparently most natural reward functions may not be optimal. They showed that even in simple problems, reward functions may exist that enable significantly faster learning than do natural task-specific reward functions. The alternate reward functions may be related to the problem in counterintuitive ways and they may require precise tuning to work well. Singh, Lewis, and Barto [6] linked these properties explicitly to evolutionary processes, noting that the discovery and tuning processes that would be required to produce optimal reward functions are akin to biological evolutionary processes. They did not, however, implement evolutionary search for reward functions; they used exhaustive search over small reward function spaces for simple problems in order to explore the nature of reward functions, but they did not seek to automate reward function search for more complex problems.

We employed evolutionary computation techniques to explore whether or not evolutionary processes could indeed be applied usefully to the search for reinforcement learning reward functions [3]. We used genetic programming (specifically PushGP, a stack-based genetic programming system built on the Push programming language [7, 8]) to search for alternate reward functions for a Q-learning system [11] in the Hungry-Thirsty domain [5]. The evolving individuals in our genetic programming system were reward functions, each of which was tested for fitness by using it to drive a Q-learning agent's behavior and learning over a "lifetime" in the problem environment. We demonstrated that this kind of evolutionary search can be worthwhile and argued that it can produce superior reward functions in a variety of circumstances, for example when an agent faces distributions of related environments, non-stationary environments, or problems in which agents have limited lifetimes. We also showed that the evolutionary search process is affected only minimally by changes to the scale of the problem being solved, and that it can continue to perform well even as the dimen-

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

sionality of the state space is increased. Details are available in our full report [3].

2. OPPORTUNITIES

Most highly adaptive natural systems involve adaptive mechanisms both at the population level (evolution) and at the individual level (learning). Experiments have been conducted for decades on various hybridizations of machine learning and evolutionary computation technologies [1], but we think that the particular hybridization that we have described here is particularly promising for several reasons.

First, the approach described here allows one to use off-the-shelf machine learning and evolutionary computation technologies. The reinforcement learning system that is used in this approach does not have to be modified at all; it just runs with a reward function that is provided by an evolutionary computation system rather than a reward function that was designed by a human. Furthermore, the approach that we have described here could be applied to any area for which reinforcement learning is applicable, and reinforcement learning has already proven to have high utility in many important application areas. Similarly, the genetic programming system does not have to be modified at all for it to be used in the framework that we have discribed; the learning all takes place within the fitness function, with the fitness of the evolving reward functions being determined by the learning performance of the simulated agent.

Second, the clean interface between the machine learning and evolutionary computation components provides great freedom, when confronting a new application area, in choosing the specific technologies to use on both sides. There are many different genetic programming techniques and many different reinforcement learning algorithms, and any of the former could conceivably be used to evolve reward functions for any of the latter. The user of our approach is free to choose the reinforcement learning technique most appropriate for the learning task that is posed by the environment, and also to choose the genetic programming technique most appropriate for the evolutionary task that is posed by the reward function fitness landscape generated by the environment in conjunction with the learning system.

Finally, we would argue that the approach that we have described here is largely consistent with at least one of the major interactions between evolution and learning in nature. Learning, not evolution, acts within an organism's lifetime, but learning is driven by in part by intrinsic motivations that are crafted by evolution. In our framework the reward functions may serve in part as intrinsic motivations, and we evolve agents that have reward functions that drive learning in ways that are advantageous to the agents and their reproductive success. Because of these parallels—which are also present in some but not all other hybridizations of machine learning and evolutionary computation technologies—and because of the extraordinary success of the natural adaptive systems that combine learning and evolution in this way, there is reason to believe that our proposed approach can produce highly successful adaptive technologies.

The approach also presents challenges, however. For example, many real-world reinforcement learning problems require fitness tests that are computationally expensive. Work in this area will therefore benefit from evolutionary computation methods that increase the speed of fitness testing [10] and techniques that allow evolutionary search to succeed with fewer fitness tests [4].

In sum, the evolution of reinforcement learning reward functions is a challenging and yet highly promising application area for evolutionary computation systems. We urge other researchers in evolutionary computation to consider further work in this area.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1017817. Andrew Barto was supported by AFOSR grant FA9550-08-1-0418. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsors.

3. REFERENCES

[1] R. K. Belew and M. Mitchell, editors. *Adaptive individuals in evolving populations: models and algorithms.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1996.

[2] A. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: theory and application to reward shaping. In *Proc. 16th Intl. Conf. on Machine Learning*, pages 278–287, 1999.

[3] S. Niekum, A. G. Barto, and L. Spector. Genetic programming for reward function search. *IEEE Trans. Autonomous Mental Development*, pages 83–90, 2010.

[4] M. D. Schmidt and H. Lipson. Co-evolving fitness predictors for accelerating and reducing evaluations. In R. L. Riolo, T. Soule, and B. Worzel, editors, *Genetic Programming Theory and Practice IV*, pages 113–130. Springer, 2006.

[5] S. Singh, R. Lewis, and A. Barto. Where do rewards come from? In *Proc. 31st Annual Conf. of the Cognitive Science Society*, pages 2601–2606, 2009.

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

[7] L. Spector, J. Klein, and M. Keijzer. The push3 execution stack and the evolution of control. In *Proc. 2005 Conf. on Genetic and Evolutionary Computation*, pages 1689–1696, 2005.

[8] L. Spector and A. Robinson. Genetic programming and autoconstructive evolution with the push programming language. *Genetic Programming and Evolvable Machines*, 3(1):7–40, Mar. 2002.

[9] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[10] M. Tomassini and L. Vanneschi. Guest editorial: special issue on parallel and distributed evolutionary algorithms, part two. *Genetic Programming and Evolvable Machines*, 11:129–130, 2010.

[11] C. Watkins. Learning from delayed rewards. *PhD Thesis University of Cambridge, England*, 1989.