INFORMS Journal on Computing
Vol. 22, No. 1, Winter 2010, pp. 18–19
ISSN 1091-9856 | EISSN 1526-5528 | 10 | 2201 | 0018

DOI 10.1287/ijoc.1090.0350

Commentary

Perspectives on Stochastic Optimization Over Time

John N. Tsitsiklis
Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, jnt@mit.edu

History: Accepted by Edward Wasil, Area Editor for Feature Articles; received December 2008; revised February 2009; accepted March 2009. Published online in *Articles in Advance* October 2, 2009.

Motivated by the discussion in Powell (2010), I offer a few comments on the interactions and merging of stochastic optimization research in artificial intelligence (AI) and operations research (OR), a process that has been ongoing for more than a decade.

In a broad sense, decision making over time and under uncertainty is a core subject in several fields that can perhaps be described collectively as the "information and decision sciences," which includes operations research, systems and control theory, and artificial intelligence. These different fields and communities have much to offer to each other.

Operations research and systems and control theory have been close for a long time. In both fields, the predominant description of uncertainty involves probabilistic models, and the goal is usually one of optimizing an objective function subject to constraints. Any differences between these two fields are due to "culture" (different departments and conferences), motivating applications (physics-based versus service-oriented systems), and technical taste (e.g., discrete versus continuous state and time), and yet the legacy of Bellman is equally strong on both sides.

AI is a little different. Originally driven by the lofty goal of understanding and reproducing "intelligence," AI involves an eclectic mix of logic, discrete mathematics, heuristics, and computation, with a focus on problems too complex to be amenable to mainstream methods such as linear or dynamic programming. Today, however, there is a notable convergence of the "modern approach" to AI (as exemplified by Russell and Norvig 1995) and the more traditional methodologies of applied mathematics. Quite often, the clever heuristic approaches developed in AI to deal with complex problems are best understood, and then enhanced, by deploying suitably adapted classical tools.

Decision making over time and under uncertainty is a prominent example of such convergence: indeed, the methods of "reinforcement learning" are best understood as methods of *approximate dynamic programming* (ADP). This connection is certainly intellectually satisfying. More important, this connection is valuable because insights and approaches developed in one field or community can (and have been) transferred to another.

A central idea connecting the two fields is the "heuristic evaluation function," initially introduced in AI game-playing programs. The ideal evaluation function, leading to optimal play, is nothing but Bellman's optimal value function, in principle computable by dynamic programming (DP) algorithms and their extensions to the context of Markov games. For difficult problems where the optimal value function is practically impossible to compute, value function approximations become useful, potentially leading to near-optimal performance. Such approximations can be developed in an ad hoc manner or through suitable approximate dynamic programming methods. The latter approach has opened up a vast range of possibilities and an active research area.

Having identified the common foundation, it is worth elaborating on some differences of emphasis in the different communities. One key distinction concerns "online" and "offline" methods. Reinforcement learning has been motivated in terms of agents that act over time, observe the consequences of their decisions, and try to improve their decision-making rule (or "policy," in DP language) on the basis of accumulated experience. (As such, reinforcement learning is also closely related to the problem of adaptive control in systems and control theory.) A typical example is provided by a poorly modeled robot operating in a poorly modeled environment that "learns" online and incrementally improves its policy and performance. Learning online is unavoidable in "model-free" problems, where an analytical or simulation model is absent.

On the other hand, most operations research applications of ADP are not of the online or model-free

type. An inventory manager who tries to learn from daily experience cannot use a learning method that requires thousands of time steps before converging to a near-optimal policy. Instead, in typical OR problems (e.g., assignment of a fleet of trucks, or airline yield management, or scheduling in a manufacturing system), a model is available, either analytically or through a simulator. This opens up the possibility of a massive offline computational effort, possibly relying on simulations of a huge number of time steps and the offline use of an online method to produce near-optimal policies. It is interesting to note that Tesauro's backgammnon player (Tesauro 1995), as well as its precursor, Samuel's checkers player (Samuel 1963), were of this type. Thus, although the popular methods of reinforcement learning (such as temporal difference methods and their relatives) were motivated by the online context, they can and are often used offline.

Online learning is by default "incremental": the information acquired at each decision epoch is used to effect a typically small change in the policy parameters. However, once we move to the offline realm, the use of an incremental method is a matter of choice, not a requirement. This leads to the possibility of batch-oriented methods. The linear programming (LP) approach to ADP (de Farias and Van Roy 2003) or the LSTD (least-squares temporal difference) (Bradtke and Barto 1996) methods are some examples where the slow convergence of iterative/learning methods of the stochastic approximation type are replaced by more efficient methods, such as the direct solution of systems of linear equations or the use of general-purpose LP solvers.

Another difference of emphasis comes from one of the central messages of AI: the choice of problem representation is important. In our context, the state of a system is often summarized by certain features or basis functions that capture the state's salient properties. The selection of suitable features is often the *condicio sine qua non* for practical success. The choice of features is almost always influenced by sound engineering understanding of the problem domain, but automating this process would be a major step ahead. Unfortunately, this is too difficult, in general, but progress is possible by, e.g., using kernel sparsification or other forms of nonparametric learning (Engel et al. 2003).

Finally, although the above discussion has centered on ways to approximate the value function, alternative approaches are also available, based on parametrizations of a limited but promising class of policies together with gradient descent in policy space, as well as on combinations of policy parametrization and value function approximation (the so-called "actor-critic" methods).

At a higher level, one important challenge is the development of streamlined methods that do not rely on the user's ingenuity. Although a general-purpose ADP package is unlikely to emerge any time soon, much useful research is possible in identifying problem classes for which particular approaches can be successfully standardized.

References

Bradtke, S. J., A. G. Barto. 1996. Linear least-squares algorithms for temporal difference learning. *Machine Learn.* **22**(1–3) 33–57.

de Farias, D. P., B. Van Roy. 2003. The linear programming approach to approximate dynamic programming. *Oper. Res.* **51**(6) 850–865.

Engel, Y., S. Mannor, R. Meir. 2003. Bayes meets Bellman: The Gaussian process approach to temporal difference learning. *Proc. 20th Internat. Conf. Machine Learn., Washington, DC.* AAAI Press, Menlo Park, CA, 154–161.

Powell, W. B. 2010. Merging AI and OR to solve high-dimensional stochastic optimization problems using approximate dynamic programming. *INFORMS J. Comput.* **22**(1) 2–17.

Russell, S. J., P. Norvig. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

Samuel, A. L. 1963. Some studies in machine learning using the game of checkers. *IBM J. Res. Development* 210–229. [Reprinted in 1995. E. A. Feigenbaum, J. Feldman, eds. *Computers and Thought*. McGraw-Hill, New York, 71–105.]

Tesauro, G. 1995. Temporal difference learning and TD-Gammon. *Commun. ACM* **38**(3) 58–68.