Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Sampled fictitious play for approximate dynamic programming

Marina Epelman [a], Archis Ghate [b,*], Robert L. Smith [a]

[a] Industrial and Operations Engineering, University of Michigan, Ann Arbor, USA
[b] Industrial and Systems Engineering, Box 352650, University of Washington, Seattle, WA 98195, USA

ARTICLE INFO

Available online 16 February 2011

Keywords:
Stochastic dynamic optimization
Computational game theory
Simulation optimization

ABSTRACT

Sampled fictitious play (SFP) is a recently proposed iterative learning mechanism for computing Nash equilibria of non-cooperative games. For games of identical interests, every limit point of the sequence of mixed strategies induced by the empirical frequencies of best response actions that players in SFP play is a Nash equilibrium. Because discrete optimization problems can be viewed as games of identical interests wherein Nash equilibria define a type of local optimum, SFP has recently been employed as a heuristic optimization algorithm with promising empirical performance. However, there have been no guarantees of convergence to a globally optimal Nash equilibrium established for any of the problem classes considered to date. In this paper, we introduce a variant of SFP and show that it converges almost surely to optimal policies in model-free, finite-horizon stochastic dynamic programs. The key idea is to view the dynamic programming states as players, whose common interest is to maximize the total multi-period expected reward starting in a fixed initial state. We also offer empirical results suggesting that our SFP variant is effective in practice for small to moderate sized model-free problems.

1. Introduction

In this paper, we introduce a variant of a game theoretic learning mechanism called sampled fictitious play (SFP) [20] to solve model-free stochastic dynamic programming problems, and investigate its convergence properties and empirical performance. The defining feature of model-free problems is that the state space, immediate rewards resulting from choosing an action in a state, and state transition probabilities are not known explicitly, and hence, system behavior must be "learned" off-line or on-line by repeated computer simulations or system runs. This rules out methods like backward induction, value iteration and mathematical programming. Examples of model-free problems include control of queueing networks with complicated service disciplines whose state transitions are available only through simulation via computer programs [3], control of manufacturing processes where the effect of a decision on the process is calculated by simulating the process [18], and dynamic portfolio optimization or financial derivative pricing problems where the performance of the underlying financial instrument is obtained by simulating complex computer models. Algorithms for model-free problems are termed "simulation based" methods [3,9,26,32] and typically provide an approximate solution. Thus, these simulation based techniques, including our SFP variant, fall within the realm of approximate dynamic programming (ADP) [26].

Stochastic search methods rooted in game theory have recently been applied to large-scale discrete optimization problems, with special focus on cases where the objective function is available only through computationally expensive simulations [2,10,14–16,20–22]. Consequently, the hope is to at least find local optima, as stronger forms of optimality are nearly impossible to attain, and very difficult to check. These techniques have been numerically tested with encouraging results on problems in transportation [10,14,22], power management in sensor networks [16], network optimization [15], and manufacturing systems [2].

Such heuristic optimization algorithms are applied to problems of the form

$$\max u(y^1,y^2,\ldots,y^n) \quad \text{s.t.} \quad (y^1,y^2\ldots,y^n) \in (Y^1\times Y^2\times\cdots\times Y^n), \tag{1}$$

where $(Y^1\times Y^2\times\cdots\times Y^n)$ denotes the Cartesian product of finite sets Y^1 through Y^n. The main idea is then to animate (1) as a game between n players (corresponding to decision variables $y^1,\ldots,y^n$), who share the *identical interest* of maximizing the objective function $u(\cdot)$. Recall that a Nash equilibrium is a collection of probability distributions over each player's actions with the property that no player can unilaterally improve its utility in expectation by changing its own distribution [13]. Such an equilibrium serves as a type of coordinate-wise local optimum of (1), and hence, the goal is to implement a computationally efficient procedure to find it.

[*] Corresponding author. Tel.: +1 206 616 5968; fax: +1 206 685 3072.
E-mail address: archis@u.washington.edu (A. Ghate).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.01.023

Most of these game theory based techniques for discrete optimization employ variants of fictitious play (FP) [6,29], a well-known iterative learning technique for computing Nash equilibria. At every iteration of FP, each player chooses a strategy that is a best response (with respect to that player's expected utility, which depends on decisions of all players) to the other players' strategies, assuming they will be chosen based on the empirical probability distribution induced by the historical frequency of their best response decisions in all previous iterations. Suitability of the FP approach for optimization stems from its convergence properties. In particular, for a game of identical interests, every limit point of the sequence of mixed strategies induced by the empirical frequencies of best response actions is a Nash equilibrium irrespective of the structure of the objective function [24]. This is termed the "fictitious play property." Another advantage of such methods is that they are easily parallelizable, making them potentially suitable for large-scale optimization problems. However, the best response problem for each player in FP is computationally intractable for realistic problems since an exact computation of the expected utility for a player may require evaluation of the utility function for every possible combination of actions for all players.

As a result, two of the authors and a co-author recently proposed sampled fictitious play (SFP) [20], a modified version of FP where the players choose an action that is a best reply to an independent *sample of actions* of other players drawn according to the empirical probability distribution of actions they used in all previous iterations. SFP offers significant computational advantage over FP, and for games of identical interests, almost surely exhibits the fictitious play property if the sample size is increased at a polynomial rate with iterations [20]. However, efficacy of the original version of SFP for optimization problems has the following significant limitations: (i) the fictitious play property guarantees convergence to only an equilibrium solution rather than an optimal solution, (ii) SFP may converge to a mixed strategy equilibrium, whereas in many applications, and especially in optimization, a pure strategy equilibrium is desirable, (iii) the best response computation becomes increasingly expensive as the sample size grows without bound with iterations, (iv) it is computationally very expensive to force every player in large-scale problems to perform a best reply computation in every iteration, and, finally, (v) problem form (1) excludes optimization problems with constraints across variables. Thus, practical implementations of SFP [2,10,21] have attempted ad hoc variations of the original version. Unfortunately, the original convergence results for SFP in [20] do not hold for these ad hoc variants. Consequently, the question as to whether one can design an SFP variant, that provably finds optimal solutions to an important class of optimization problems by surmounting the above difficulties, has remained open. We answer this question in the affirmative by introducing an SFP variant that solves finite-horizon stochastic dynamic programs.

The key idea is to view the states of the dynamic programming problem as players engaged in a non-cooperative game of *identical interests*, where the objective of each player is to maximize the expected multi-period reward from the *initial state*. The problem structure inherent in dynamic programming, and specifically, the principle of optimality, help our SFP players coordinate their actions and hence solve the problem to optimality. Viewing the states as players also has the important advantage that all combinations of feasible actions of these players are jointly feasible so that the resulting problem is an unconstrained one of the form (1). Importantly, since the objectives of all players are aligned with one another, it suffices for a very small fraction of the players to participate in the game in each iteration, naturally leading to an asynchronous procedure. The procedure to determine which players will participate in an iteration adaptively favors optimal actions (and hence the states they deliver) from the recent past. Specifically, unlike the original SFP in [20], we provide the players with only finite memory. Moreover, we allow players to sample only one action at a time (unlike the original SFP version in [20] which requires an increasing action sample size), and we deliberately add exogenous noise to this selection procedure so that every player in theory gets an opportunity to perform a best response computation infinitely often with probability one even in the asynchronous case. We also remark that if the inherent randomness in state transitions of the system is such that all states will be observed infinitely often with probability one irrespective of the policy implemented, then this exogenous noise is not needed (even in the asynchronous case).

This paper is organized as follows. We develop the necessary notation and formulate our dynamic programming problem in the second section. This is followed by a precise description of our SFP algorithm in the third section. Convergence results and proofs appear in the fourth section. Numerical experiments are presented in the fifth section. Since our SFP variant falls within the realm of simulation-based algorithms for approximate dynamic programming (ADP) [26], a detailed discussion of similarities and differences between our approach and existing simulation-based techniques for ADP is provided in the sixth section, along with other specific conclusions, and future research directions.

2. Problem formulation

Even though our SFP approach is applicable to any finite-state, finite-action, finite-horizon stochastic dynamic program, it is presented here for the case of model-free problems [3,26]. In particular, consider the following T period sequential decision problem. The initial state of a system is s_1. In each period $t=1,2,\ldots,T$, we observe state s_t of the system and make a decision x_t, which must be chosen from a finite set $X_t(s_t)$ of feasible decisions in state s_t, as determined by a feasibility oracle $\mathcal{F}_t$. This feasibility oracle receives s_t as input and produces a finite set $X_t(s_t)$ as output. Then, another oracle $\mathcal{O}_t$ receives s_t and x_t as input and returns the (stochastic) state of the system s_{t+1} in the next period, and a one period deterministic reward $r_t(s_t,x_t)$. It is common in the literature to consider deterministic one period rewards [7,33]. The slightly more general case of random one period rewards can also be handled by our algorithm, and the convergence results in Section 4 can be extended to that case without much difficulty. All states of the form s_{T+1} are "terminal" states and there are no feasible actions in these states. We adopt the convention that terminal states have no intrinsic value. Our results generalize in a straightforward manner to problems where nonzero values are assigned to terminal states as for example in our TIC-TAC-TOE example in Section 5. We use S_t to denote the set of feasible states at the beginning of period t, $t=1,2,\ldots,T+1$ with $S_1=\{s_1\}$. Let S denote the finite set of all feasible states of the system, i.e., $S=S_1\cup S_2\ldots\cup S_{T+1}$. The sets $S_2, S_3,\ldots,S_{T+1}$, and (hence) the state space S are not known a priori, but must be "discovered" by repeatedly querying oracles $\mathcal{F}_t$ and $\mathcal{O}_t$.

A policy π is a deterministic decision rule that assigns a feasible decision x_t to every state s_t in S. Thus, π is a $|S|$-dimensional vector, and the decision that policy π assigns to state $s_t\in S$ is denoted by $\pi(s_t)$. We use Π to denote the set of all feasible policies of this form. Therefore, for each $s_t\in S$, any policy $\pi\in\Pi$ must have the property that $\pi(s_t)\in X_t(s_t)$. Let $V_\pi(s_1)$ be the value of state s_1 under policy π, i.e., the total T-period expected reward from the initial state if we implement decisions prescribed by policy π in every