Results in Control and Optimization 1 (2020) 100003

Contents lists available at ScienceDirect

Results in Control and Optimization

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

Multiagent value iteration algorithms in dynamic programming and reinforcement learning

Dimitri Bertsekas *

McAfee Professor of Engineering, MIT, Cambridge, MA, United States of America
Fulton Professor of Computational Decision Making, ASU, Tempe, AZ, United States of America

A B S T R A C T

We consider infinite horizon dynamic programming problems, where the control at each stage consists of several distinct decisions, each one made by one of several agents. In an earlier work we introduced a policy iteration algorithm, where the policy improvement is done one-agent-at-a-time in a given order, with knowledge of the choices of the preceding agents in the order. As a result, the amount of computation for each policy improvement grows linearly with the number of agents, as opposed to exponentially for the standard all-agents-at-once method. For the case of a finite-state discounted problem, we showed convergence to an agent-by-agent optimal policy. In this paper, this result is extended to value iteration and optimistic versions of policy iteration, as well as to more general DP problems where the Bellman operator is a contraction mapping, such as stochastic shortest path problems with all policies being proper.

1. Multiagent problem formulation

We consider an abstract form of infinite horizon dynamic programming (DP) problem, which contains as special case finite-state discounted Markovian decision problems (MDP), as well as more general problems where the Bellman operator is a monotone weighted sup-norm contraction. The distinguishing feature of the problem is that the control u consists of m components u_ℓ, $\ell = 1, \dots, m$, where $m > 1$:

$$u = (u_1, \dots, u_m). \tag{1.1}$$

Conceptually, each component may be viewed as chosen by a distinct agent, with knowledge of the selections of the other agents. We consider value iteration (VI) algorithms that involve minimization component-by-component as opposed to minimization over all components at once. This is similar to what is done in coordinate descent methods for multivariable optimization, and can lead to dramatic gains in computational efficiency for large and even moderate values of m. We propose several methods and we show their convergence to an agent-by-agent optimal policy, a type of policy that is related to the notion of person-by-person optimality from the theory of teams. Our analysis extends and complements our earlier proposals of rollout and policy iteration (PI) algorithms [1,2].

We assume that u is chosen from a finite constraint set $U(x)$ when the system is at state x. In our earlier papers [1,2], we have made a stronger assumption: we assumed that each control component u_ℓ, $\ell = 1, \dots, m$, is separately constrained to lie in a given finite set $U_\ell(x)$. In this case $U(x)$ is the Cartesian product set

$$U(x) = U_1(x) \times \cdots \times U_m(x). \tag{1.2}$$

In this paper, we do not impose this assumption, except occasionally to discuss its implications. As a result our algorithms must ensure that the selection of a control component at a given state and stage does not preclude the feasibility of selection of the other control components at the same state and stage. This complicates our algorithms relative to the Cartesian product case (1.2). We will discuss the mechanism for dealing with this issue in Section 2. For the remainder of this section, we will assume no special structure for the constraint set $U(x)$ other than finiteness.

* Correspondence to: Fulton Professor of Computational Decision Making, ASU, Tempe, AZ, United States of America
E-mail address: dbertsek@asu.edu.

https://doi.org/10.1016/j.rico.2020.100003
Received 6 October 2020; Accepted 1 November 2020
Available online 10 November 2020

The α-discounted MDP case

A major context for application of our algorithmic ideas is the standard infinite horizon discounted MDP with states $x = 1, \dots, n$. Here, at state x, a control u is applied, and the system transitions to a next state y with transition probabilities $p_{xy}(u)$ and cost $g(x, u, y)$. The control is chosen at state x from a finite constraint set $U(x)$. The cost function of a stationary policy μ that applies control $\mu(x) \in U(x)$ at state x is denoted by $J_\mu(x)$, and the optimal cost [the minimum over μ of $J_\mu(x)$] is denoted by $J^*(x)$.

The standard VI algorithm starts from some initial guess J^0 and iterates as follows[1]:

$$J^{k+1} = TJ^k, \qquad k = 0, 1, \dots,$$

where T is the Bellman operator, which maps a vector $J = \big(J(1), \dots, J(n)\big)$ to the vector

$$TJ = \big((TJ)(1), \dots, (TJ)(n)\big)$$

according to

$$(TJ)(x) = \min_{u \in U(x)} \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha J(y)\big), \qquad x = 1, \dots, n. \tag{1.3}$$

Thus each VI involves a comparison of all the Q-factors

$$Q(x, u) = \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha J(y)\big), \qquad x = 1, \dots, n,\ u \in U(x). \tag{1.4}$$

A related algorithm is *optimistic PI*, which involves simultaneous value and policy iterations, using the Bellman operator T_μ defined for each policy μ by

$$(T_\mu J)(x) = \sum_{y=1}^{n} p_{xy}\big(\mu(x)\big)\Big(g\big(x, \mu(x), y\big) + \alpha J(y)\Big), \qquad x = 1, \dots, n. \tag{1.5}$$

Given a pair (μ^k, J^k), this algorithm generates (μ^{k+1}, J^{k+1}) according to

$$T_{\mu^{k+1}} J^k = TJ^k, \qquad J^{k+1} = T^q_{\mu^{k+1}} J^k, \qquad k = 0, 1, \dots, \tag{1.6}$$

where q is a positive integer (which in some cases may depend on k), and T^q_μ denotes the mapping obtained by q-fold application of the mapping T_μ. When $q = 1$ we obtain the VI algorithm $J^{k+1} = TJ^k$ and when $q \to \infty$, we have $J^{k+1} = J_{\mu^k}$ (in the limit), so the algorithm approaches the standard PI algorithm where μ^{k+1} is obtained from μ^k according to

$$T_{\mu^{k+1}} J_{\mu^k} = TJ_{\mu^k}. \tag{1.7}$$

Unfortunately, iterating with the mapping T is inconvenient for problems involving even a moderate number of agents, because the size of the control constraint set $U(x)$ typically grows exponentially with m. In particular, in the Cartesian product case (1.2), if each constraint set $U_\ell(x)$ consists of at most s elements, minimization over $U(x)$ involves a comparison of as many as s^m Q-factors of the form (1.4). This motivates us to consider versions of the preceding algorithms that involve a simpler form of minimization. For example, minimization over the component constraint sets $U_\ell(x)$, one component at a time, which involves a comparison of s Q-factors for each agent, for a total of $s \cdot m$ Q-factors.

The general contractive DP case

It is convenient and useful to develop our algorithm in a more general setting, which involves an operator-based framework from the author's abstract DP book [3]. In particular, we consider *a finite set X of states and a finite set U of controls*, and for each $x \in X$, a nonempty control constraint set $U(x) \subset U$.[2] We denote by $\mathcal{M}$ the set of all functions $\mu : X \mapsto U$ with $\mu(x) \in U(x)$ for all $x \in X$, which we refer to as *policies*. We introduce a mapping $H : X \times U \times \mathcal{R}(X) \mapsto \Re$, where $\Re$ denotes the real line and $\mathcal{R}(X)$ denotes the set of real-valued functions $J : X \mapsto \Re$. For each policy $\mu \in \mathcal{M}$, we consider the mapping $T_\mu : \mathcal{R}(X) \mapsto \mathcal{R}(X)$ defined by

$$(T_\mu J)(x) = H\big(x, \mu(x), J\big), \qquad x \in X.$$

We also consider the mapping T defined by

$$(TJ)(x) = \min_{u \in U(x)} H(x, u, J) = \min_{\mu \in \mathcal{M}} (T_\mu J)(x), \qquad x \in X.$$

Note that the α-discounted MDP is obtained when H is given by

$$H(x, u, J) = \sum_{y=1}^{n} p_{xy}(u)\big(g(x, u, y) + \alpha J(y)\big), \qquad x = 1, \dots, n. \tag{1.8}$$

[1] Throughout the paper, we will be using componentwise equality and inequality notation, whereby for any pair of real-valued functions J, J' of the state x, we write $J = J'$ (or $J \le J'$) if $J(x) = J'(x)$ [or $J(x) \le J'(x)$, respectively] for all x.

[2] The abstract DP framework of [3] does not require finiteness of the state and control spaces. We impose the finiteness assumption in order to obtain the most powerful algorithmic results possible. However, at several points in the paper, and particularly in Section 5, we speculate around the possibility of extending our algorithms and analysis to infinite state and control spaces.

The problem is to find a function $J^* \in \mathcal{R}(X)$ such that

$$J^*(x) = \min_{u \in U(x)} H(x, u, J^*) = (TJ^*)(x), \qquad x \in X,$$

i.e., to find a fixed point of T within $\mathcal{R}(X)$ (we can view $J^* = TJ^*$ as a generalized form of Bellman's equation). We also want to obtain a policy $\mu^* \in \mathcal{M}$ such that $T_{\mu^*} J^* = TJ^*$. We assume that the control u consists of the m components u_ℓ, $\ell = 1, \ldots, m$, [cf. Eq. (1.1)]. Note that since the state and control spaces are assumed finite, the control constraint set $U(x)$ and the set of policies $\mathcal{M}$ are also finite, so the minimum of various expressions over $U(x)$ or $\mathcal{M}$ is attained.

We will adopt throughout the following monotonicity and contraction assumptions.

Assumption 1.1 (*Monotonicity*)**.** If $J, J' \in \mathcal{R}(X)$ and $J \leq J'$, then

$$H(x, u, J) \leq H(x, u, J'), \qquad \forall\ x \in X,\ u \in U(x).$$

For the contraction assumption, we introduce a function $v : X \mapsto \Re$ with

$$v(x) > 0, \qquad \forall\ x \in X.$$

We consider the weighted sup-norm

$$\|J\| = \max_{x \in X} \frac{\left|J(x)\right|}{v(x)}$$

on $\mathcal{R}(X)$, the space of real-valued functions J on X.

Assumption 1.2 (*Contraction*)**.** For some $\alpha \in (0, 1)$, we have

$$\|T_\mu J - T_\mu J'\| \leq \alpha \|J - J'\|, \qquad \forall\ J, J' \in \mathcal{R}(X),\ \mu \in \mathcal{M}.$$

The monotonicity and contraction assumptions are satisfied in the α-discounted finite-state MDP case (1.8), as well as other finite-state DP problems, such as stochastic shortest path problems in the special case where all policies are proper; see the books [3–5] for an extensive discussion. In particular, for the α-discounted MDP, T_μ is a contraction with respect to the unweighted sup-norm with contraction modulus α, whereas in the stochastic shortest path case, T_μ is a contraction with respect to a weighted sup-norm with weights and contraction modulus that depend on the maximum expected time to reach the destination using proper policies (see [4], Prop. 2.2).

General abstract DP models under Assumptions 1.1 and 1.2 have been investigated in detail in the author's monograph [3] [without assuming finiteness of X and U, but with $\mathcal{R}(X)$ replaced by the set $\mathcal{B}(X)$ of all uniformly bounded functions over X, equipped with a weighted sup-norm]. The main results are that T is a contraction mapping and has as unique fixed point the optimal cost function J^* (the equation $J^* = TJ^*$ is Bellman's equation). Also J_μ is the unique fixed point of T_μ. Moreover μ is optimal if and only if $T_\mu J^* = TJ^*$ (or equivalently $T_\mu J_\mu = TJ_\mu$). Algorithmic results include the convergence of VI [i.e., $T^k J \to J^*$ for all $J \in \mathcal{R}(X)$], and also convergence results for the PI algorithm (1.7) and some of its variations. We will be using these results in what follows in this paper, with the monograph [3] as a general reference. For parts of our analysis, only the monotonicity and contraction Assumptions 1.1 and 1.2 are essential: the assumption of finiteness of the state and control spaces can be eliminated with minor mathematical proof modifications.

2. Agent-by-agent value iteration

The salient feature of the multiagent DP problem of this paper is that the control u consists of m components,

$$u = (u_1, \ldots, u_m);$$

cf. Eq. (1.1). We will aim to develop a computationally efficient variant of the standard VI algorithm $J^{k+1} = TJ^k$, i.e.,

$$J^{k+1}(x) = \min_{(u_1, \ldots, u_m) \in U(x)} H(x, u_1, \ldots, u_m, J^k), \qquad x \in X.$$

Rather than simultaneous minimization over all the components $u_1, \ldots, u_m$, *our multiagent VI algorithm involves sequential minimization of $H(x, u_1, \ldots, u_m, J^k)$ over a single component u_ℓ, with the remaining components $u_{\ell'}$, $\ell' \neq \ell$, fixed at the values obtained through the preceding minimizations*. We maintain these control component values in a policy that is continually updated to incorporate the results of new minimizations.

Let μ be a given policy that applies at x the control

$$\mu(x) = \big(\mu_1(x), \ldots, \mu_m(x)\big).$$

We define a constraint set for the ℓth control component u_ℓ that is given by

$$U_{\ell,\mu}(x) = \Big\{ u_\ell \mid \big(\mu_1(x), \ldots, \mu_{\ell-1}(x), u_\ell, \mu_{\ell+1}(x), \ldots, \mu_m(x)\big) \in U(x) \Big\}, \qquad \ell = 1, \ldots, m. \tag{2.1}$$

Note that since a policy μ by definition satisfies the feasibility constraint

$$\big(\mu_1(x), \ldots, \mu_m(x)\big) \in U(x), \qquad x \in X,$$

the set $U_{\ell,\mu}(x)$ contains $\mu_\ell(x)$, so it is nonempty. Note also that when $U(x)$ has the Cartesian product form $U_1(x)\times\cdots\times U_m(x)$, the set $U_{\ell,\mu}(x)$ is simply equal to $U_\ell(x)$ for all μ.

Our algorithm generates a double sequence $\{J^k,\mu^k\}$, starting from some pair (J^0,μ^0): at the kth iteration, given (J^k,μ^k), the algorithm obtains (J^{k+1},μ^{k+1}) after m successive minimizations, one for each of the components u_ℓ, $\ell=1,\ldots,m$. In particular, given the typical pair (J,μ), our algorithm generates the next pair $(\tilde J,\tilde\mu)$ as the last of a sequence of cost function-policy component pairs

$$(\hat J_1,\hat\mu_1),(\hat J_2,\hat\mu_2),\ldots,(\hat J_m,\hat\mu_m), \tag{2.2}$$

to be defined shortly, i.e., it sets

$$\tilde J(x)=\hat J_m(x),\qquad \tilde\mu(x)=\big(\hat\mu_1(x),\ldots,\hat\mu_m(x)\big),\qquad x\in X. \tag{2.3}$$

The cost function-policy pairs (2.2) are obtained as follows:

For every $\ell=1,\ldots,m$, given $(\hat J_{\ell-1},\hat\mu_1,\ldots,\hat\mu_{\ell-1})$, the algorithm generates $(\hat J_\ell,\hat\mu_\ell)$ according to

$$\hat J_\ell(x)=\min_{u_\ell\in U_{\ell,(\hat\mu_1,\ldots,\hat\mu_{\ell-1},\mu_\ell,\ldots,\mu_m)}(x)} H\big(x,\hat\mu_1(x),\ldots,\hat\mu_{\ell-1}(x),u_\ell,\mu_{\ell+1}(x),\ldots,\mu_m(x),\hat J_{\ell-1}\big),\qquad x\in X, \tag{2.4}$$

$$\hat\mu_\ell(x)\in\operatorname*{argmin}_{u_\ell\in U_{\ell,(\hat\mu_1,\ldots,\hat\mu_{\ell-1},\mu_\ell,\ldots,\mu_m)}(x)} H\big(x,\hat\mu_1(x),\ldots,\hat\mu_{\ell-1}(x),u_\ell,\mu_{\ell+1}(x),\ldots,\mu_m(x),\hat J_{\ell-1}\big),\qquad x\in X, \tag{2.5}$$

where the constraint set in the two preceding minimizations,

$$U_{\ell,(\hat\mu_1,\ldots,\hat\mu_{\ell-1},\mu_\ell,\ldots,\mu_m)}(x),$$

is defined by Eq. (2.1); it is the set of u_ℓ, which are consistent (in terms of feasibility) with the previously chosen components $\hat\mu_1(x),\ldots,\hat\mu_{\ell-1}(x)$ and the component choices $\mu_{\ell+1}(x),\ldots,\mu_m(x)$ specified by the policy μ. To start this process, only the initial function $\hat J_0$ is needed (in addition to μ), and it is given by

$$\hat J_0(x)=J(x),\qquad x\in X. \tag{2.6}$$

Note that each of the minimizations (2.4) is performed for every state $x\in X$, and that there may be multiple possible policies $\tilde\mu$ that can be generated by this process [cf. Eq. (2.3)], since the minimum in Eq. (2.5) may not be uniquely attained. Similarly, there may be multiple possible functions $\tilde J$ that can be generated by this process [since the minimization (2.4) is affected by the multiplicity of possible policies $\hat\mu_1,\ldots,\hat\mu_{\ell-1}$]. In summary, our multiagent VI algorithm, starting from the pair (J^k,μ^k), generates the pair (J^{k+1},μ^{k+1}) according to

$$(J^{k+1},\mu^{k+1})\in\widetilde{\mathcal{P}}(J^k,\mu^k), \tag{2.7}$$

where $\widetilde{\mathcal{P}}(J^k,\mu^k)$ is the set of cost function-policy pairs $(\tilde J,\tilde\mu)$ that can be generated by the process (2.3)–(2.6), starting with $J=J^k$ and $\mu=\mu^k$.

Optimistic and asynchronous PI algorithms

In the preceding algorithm (2.7), each iteration involves a policy improvement operation, i.e., an m-step minimization that cycles through all control components one-by-one. In Section 3, we will also consider an optimistic PI variant where the m-step minimization is performed for only an infinite subset $\mathcal{K}\subset\{0,1,\ldots\}$ of the iterations, while for the complementary subset of iterations, $k\notin\mathcal{K}$, we use the (less expensive) standard policy evaluation update $J^{k+1}=T_{\mu^k}J^k$, and no policy update:

$$(J^{k+1},\mu^{k+1})\in\widetilde{\mathcal{P}}(J^k,\mu^k),\qquad \forall\ k\in\mathcal{K}, \tag{2.8}$$

$$J^{k+1}=T_{\mu^k}J^k,\qquad \mu^{k+1}=\mu^k,\qquad \forall\ k\notin\mathcal{K}. \tag{2.9}$$

We call the algorithm (2.8)–(2.9) *multiagent optimistic PI*. It is a natural multiagent extension of the standard (single agent) optimistic PI algorithm (1.6), which is described in many sources for MDP and other problems, e.g., [3,5,6]. Note that when $\mathcal{K}=\{0,1,\ldots\}$, the optimistic PI algorithm is the same as the multiagent VI algorithm (2.7). In cases where $\mathcal{K}$ is a "small" subset of $\{0,1,\ldots\}$, the multiagent optimistic PI algorithm involves nearly exact policy evaluations and "approaches" the multiagent PI algorithm proposed in our earlier papers [1,2].

In the preceding multiagent algorithms (2.7) and (2.8)–(2.9), the iterations are performed simultaneously for all states $x\in X$. In Section 4, we will also consider an asynchronous distributed version of the multiagent optimistic PI algorithm (2.8)–(2.9), whereby iteration k is performed for only a subset X_k of the states. A requirement here is that each state x belongs infinitely often to some subset X_k, so that there are infinitely many policy improvements at every state. This algorithm is well suited for distributed asynchronous computation, involving a partition of the state space into subsets, and with a processor assigned to each set of the partition.

3. Convergence to an agent-by-agent optimal policy

We will prove that the multiagent VI algorithm (2.7) converges to an agent-by-agent optimal policy, which we define as follows.

Definition 3.1 (*Agent-by-Agent Optimality*)**.** We say that a policy $\mu = \{\mu_1, \dots, \mu_m\}$ is *agent-by-agent optimal* if for all $x \in X$ and $\ell = 1, \dots, m$, we have

$$H\big(x, \mu_1(x), \dots, \mu_m(x), J_\mu\big) = \min_{u_\ell \in U_{\ell,\mu}(x)} H\big(x, \mu_1(x), \dots, \mu_{\ell-1}(x), u_\ell, \mu_{\ell+1}(x), \dots, \mu_m(x), J_\mu\big), \tag{3.1}$$

where the constraint set $U_{\ell,\mu}(x)$ is defined by Eq. (2.1).

To interpret this definition, let a policy $\mu = \{\mu_1, \dots, \mu_m\}$ be given, and consider for every $\ell = 1, \dots, m$ the single agent DP problem where for all $\ell' \neq \ell$ the ℓ'th policy components are fixed at $\mu_{\ell'}$, while the ℓth policy component is subject to optimization. The Definition 3.1 is the optimality condition for all the single agent problems; see [3], Chapter 2 [Eq. (3.1) can be written as $T_{\mu,\ell} J_\mu = T_\ell J_\mu$, where T_ℓ and $T_{\mu,\ell}$ are the Bellman operators (1.3) and (1.5) that correspond to the single agent problem involving agent ℓ]. We can then conclude that $\mu = \{\mu_1, \dots, \mu_m\}$ is agent-by-agent optimal if each component μ_ℓ is optimal for the ℓth single agent problem, where it is assumed that the remaining policy components remain fixed; in other words by using μ_ℓ, each agent ℓ acts optimally, assuming all other agents $\ell' \neq \ell$ continue to use the corresponding policy components $\mu_{\ell'}$.

Our definition of an agent-by-agent optimal policy is related to the notion of "person-by-person" optimality from team theory, which has been studied primarily in the context of multiagent decision problems with nonclassical information patterns, whereby the agents may not share the information on which they base their decision. Thus team problems do not assume the shared information pattern that is characteristic of DP problems. For the origins of team theory and control with a nonclassical information pattern, we refer to Marschak [7], Radner [8], and Witsenhausen [9–11]. For a sampling of subsequent works, we refer to the survey by Ho [12], and the papers by Krainak, Speyer, and Marcus [13,14], de Waal and van Schuppen [15]. For more recent works, see Nayyar, Mahajan, and Teneketzis [16], Nayyar and Teneketzis [17], Li et al. [18], Gupta [19], the book by Zoppoli, Sanguineti, Gnecco, and Parisini [20], and the references quoted there.

Note that an (overall) optimal policy is agent-by-agent optimal, but the reverse may not be true. This is similar to properties of person-by-person optimal solutions in team theory. It is also similar to what may happen in coordinate descent methods for multivariable optimization, where it is possible (in the absence of favorable assumptions) to stop at a nonoptimal point where no progress can be made along any one coordinate; some examples involving a Cartesian product constraint set of the form (1.2) are given in the papers [1,2].

While an agent-by-agent optimal policy may be either optimal or adequate for practical purposes, it may offer no guarantees of quality. For a simple example, let $U(x)$ be the intersection of a Cartesian product of finite subsets $U_\ell(x)$ of the real line and the unit simplex:

$$U(x) = \big\{U_1(x) \times \cdots \times U_m(x)\big\} \cap \{u \mid u_1 + \cdots + u_m = 1\}.$$

Then it can be seen that the constraint set $U_{\ell,\mu}(x)$ consists of just the single point $\mu_\ell(x)$, so that *all* feasible policies are agent-by-agent optimal. This is due to the extreme coupling of the control components through the simplex constraint. It would not happen if the constraint set was just a Cartesian product $U_1(x) \times \cdots \times U_m(x)$, in which case $U_{\ell,\mu}(x) = U_\ell(x)$ for all ℓ. Nonetheless, one should be aware that the method of partitioning of the control into components may seriously impact the effectiveness of our multiagent VI algorithm through the creation of spurious agent-by-agent optimal policies.

We will now prove our main convergence result, under the following assumption, which is reminiscent of strict convexity assumptions in the analysis of coordinate descent methods (see e.g., [21], Section 3.7). While we do not have a concrete counterexample, we speculate based on experience with coordinate descent methods, that the assumption cannot be easily dispensed with.

Assumption 3.1 (*Uniqueness Property*)**.** The cost functions of distinct policies are distinct, i.e., for any two policies μ and μ'

$$\mu \neq \mu' \qquad \Longrightarrow \qquad J_\mu \neq J_{\mu'}.$$

Our convergence result also assumes that the initial condition (J^0, μ^0) satisfies

$$T_{\mu^0} J^0 \leq J^0. \tag{3.2}$$

This assumption is unnecessary for the α-discounted MDP where T and T_μ are given by Eqs. (1.3) and (1.5). The reason is that if we replace J^0 by a function $\overline{J}^0$ obtained by shifting J^0 by a constant c [i.e., replace $J^0(x)$ by $J^0(x) + c$ for all x], we will have

$$(T_{\mu^0} \overline{J}^0)(x) = (T_{\mu^0} J^0)(x) + \alpha c \leq J^0(x) + c = \overline{J}^0(x),$$

provided c is large enough, thereby satisfying the assumption (3.2). At the same time, it can be seen that by replacing J^0 with $\overline{J}^0$ the generated policies will not be affected, while the generated iterates J^k will just be shifted by an appropriate constant. Thus for discounted MDP the assumption (3.2) is unnecessary for the following convergence result, since the same sequence of policies will be obtained whether we use J^0 or $\overline{J}^0$.

For other types of problems the assumption (3.2) is needed. However, thanks to the contraction property of Assumption 1.2 , it can be typically satisfied by adding to $J^0(x)$ a sufficiently large constant c for all x. In particular, any function $\overline{J}$ that satisfies

$$\alpha \|\overline{J} - J_\mu\| \leq \frac{\overline{J}(x)}{v(x)} - \frac{J_\mu(x)}{v(x)}, \qquad \forall\, x \in X,\ \mu \in \mathcal{M}, \tag{3.3}$$

(for example a sufficiently large constant function) also satisfies the condition (3.2). To see this, note that for $J \leq \overline{J}$ and $x \in X$,

$$\frac{(T_\mu J)(x)}{v(x)} \leq \frac{(T_\mu \overline{J})(x)}{v(x)} \leq \frac{(T_\mu J_\mu)(x)}{v(x)} + \alpha \|\overline{J} - J_\mu\| = \frac{J_\mu(x)}{v(x)} + \alpha \|\overline{J} - J_\mu\| \leq \frac{\overline{J}(x)}{v(x)},$$

where the first inequality follows from the monotonicity of H, the second inequality follows by applying the contraction property with $J = \overline{J}$, $J' = J_\mu$, and the third inequality is Eq. (3.3). Thus, for $\overline{J}$ satisfying Eq. (3.3), we have $T_\mu \overline{J} \leq \overline{J}$ for all $\mu \in \mathcal{M}$.

Proposition 3.1 (*VI Convergence to an Agent-by-Agent Optimal Policy*)**.** *Let Assumptions 1.1, 1.2, and 3.1 hold, and assume further that the state and control spaces X and U are finite, and that the initial pair (J^0, μ^0) satisfies Eq. (3.2). Let $\{J^k, \mu^k\}$ be a sequence generated by the agent-by-agent VI algorithm (2.7). Then there is an agent-by-agent optimal policy $\bar{\mu}$ and an index $\overline{k}$ such that for all $k \geq \overline{k}$, we have $\mu^k = \bar{\mu}$, and*

$$\|J^{k+1} - J_{\bar{\mu}}\| \leq \alpha \|J^k - J_{\bar{\mu}}\|, \tag{3.4}$$

while the sequence $\{J^k\}$ converges to $J_{\bar{\mu}}$.

Proof. The critical step of the proof is to show that for all (J, μ) with

$$T_\mu J \leq J,$$

and all $(\tilde{J}, \tilde{\mu}) \in \widetilde{\mathcal{P}}(J, \mu)$ [cf. Eq. (2.7)], the following monotone decrease inequality holds

$$T_{\tilde{\mu}} \tilde{J} \leq \tilde{J} = \hat{J}_m \leq \hat{J}_{m-1} \leq \hat{J}_{m-2} \leq \cdots \leq \hat{J}_1 \leq T_\mu J \leq J, \tag{3.5}$$

where for all $\ell = 1, \ldots, m$, and $x \in X$,

$$\begin{aligned} \hat{J}_\ell(x) &= \big(T_{(\hat{\mu}_1, \ldots, \hat{\mu}_\ell, \mu_{\ell+1}, \ldots, \mu_m)} \hat{J}_{\ell-1}\big)(x) \\ &= \min_{u_\ell \in U_{\ell,(\hat{\mu}_1, \ldots, \hat{\mu}_{\ell-1}, \mu_\ell, \ldots, \mu_m)}(x)} H\big(x, \hat{\mu}_1(x), \ldots, \hat{\mu}_{\ell-1}(x), u_\ell, \mu_{\ell+1}(x), \ldots, \mu_m(x), \hat{J}_{\ell-1}\big), \end{aligned} \tag{3.6}$$

with $\hat{J}_0 = J$ [cf. Eq. (2.4)], and

$$\hat{\mu}_\ell(x) \in \operatorname*{argmin}_{u_\ell \in U_{\ell,(\hat{\mu}_1, \ldots, \hat{\mu}_{\ell-1}, \mu_\ell, \ldots, \mu_m)}(x)} H\big(x, \hat{\mu}_1(x), \ldots, \hat{\mu}_{\ell-1}(x), u_\ell, \mu_{\ell+1}(x), \ldots, \mu_m(x), \hat{J}_{\ell-1}\big),$$

[cf. Eq. (2.5)]. Indeed the relation (3.5) is proved starting from the right side, which is the assumption (3.2), and by using the definition of the algorithm, and the monotonicity Assumption 1.1 to prove first that $\hat{J}_1 \leq T_\mu J \leq J$, and then by proceeding sequentially to the inequality $\hat{J}_m \leq \hat{J}_{m-1}$. In particular, at the typical step, assuming that $\hat{J}_{\ell-1} \leq \hat{J}_{\ell-2}$, we show that $\hat{J}_\ell \leq \hat{J}_{\ell-1}$ by writing

$$\begin{aligned} \hat{J}_{\ell-1}(x) &= H\big(x, \hat{\mu}_1(x), \ldots, \hat{\mu}_{\ell-1}(x), \mu_\ell(x), \mu_{\ell+1}(x), \ldots, \mu_m(x), \hat{J}_{\ell-2}\big), \\ &\geq H\big(x, \hat{\mu}_1(x), \ldots, \hat{\mu}_{\ell-1}(x), \mu_\ell(x), \mu_{\ell+1}(x), \ldots, \mu_m(x), \hat{J}_{\ell-1}\big), \\ &\geq \min_{u_\ell \in U_{\ell,(\hat{\mu}_1, \ldots, \hat{\mu}_{\ell-1}, \mu_\ell, \ldots, \mu_m)}(x)} H\big(x, \hat{\mu}_1(x), \ldots, \hat{\mu}_{\ell-1}(x), u_\ell, \mu_{\ell+1}(x), \ldots, \mu_m(x), \hat{J}_{\ell-1}\big), \\ &= \hat{J}_\ell(x), \end{aligned}$$

where the first inequality follows by using the monotonicity Assumption 1.1 and the hypothesis $\hat{J}_{\ell-1} \leq \hat{J}_{\ell-2}$. Finally, by applying $T_{\tilde{\mu}}$ to the relation $\hat{J}_m \leq \hat{J}_{m-1}$ to obtain $T_{\tilde{\mu}} \hat{J}_m \leq T_{\tilde{\mu}} \hat{J}_{m-1}$, and by using the facts $\tilde{J} = \hat{J}_m = T_{\tilde{\mu}} \hat{J}_{m-1}$, we obtain the leftmost relation $T_{\tilde{\mu}} \tilde{J} \leq \tilde{J} = \hat{J}_m$ in Eq. (3.5). (Note that the contraction assumption is not needed for the preceding argument, and this is useful for applying this line of proof in other DP problem contexts.)

From Eq. (3.5), we see that the sequence of functions J^k converges monotonically to some function $\overline{J}$, and the same is true for all the sequences of intermediate functions $J_1^k, \ldots, J_{m-1}^k$. For each ℓ, let the policies

$$(\mu_1^{k+1}, \ldots, \mu_\ell^{k+1}, \mu_{\ell+1}^k, \ldots, \mu_m^k)$$

be equal to some policy $\bar{\mu}[\ell] = \big(\bar{\mu}_1[\ell], \ldots, \bar{\mu}_m[\ell]\big)$ infinitely often, say for an infinite index set $\mathcal{K}_\ell$, (such a policy exists since the set of policies is finite). Then we will have for all $x \in X$ and $\ell = 1, \ldots, m$,

$$J_\ell^k(x) = H(x, \bar{\mu}[\ell](x), J_{\ell-1}^k) = \min_{u_\ell \in U_{\ell,\bar{\mu}[\ell]}(x)} H\big(x, \bar{\mu}_1[\ell](x), \ldots, \bar{\mu}_{\ell-1}[\ell](x), u_\ell, \bar{\mu}_{\ell+1}[\ell](x), \ldots, \bar{\mu}_m[\ell](x), J_{\ell-1}^k\big),$$

for all $k \in \mathcal{K}_\ell$. By taking limit as $k \to \infty$, $k \in \mathcal{K}_\ell$, and using the continuity of $H(x, u, \cdot)$ (which is implied by the contraction property of $T_{\bar{\mu}[\ell]}$), we have

$$\overline{J}(x) = H\big(x, \bar{\mu}[\ell](x), \overline{J}\big) = (T_{\bar{\mu}[\ell]} \overline{J})(x), \qquad \ell = 1, \ldots, m, \ x \in X, \tag{3.7}$$

as well as

$$\overline{J}(x) = \min_{u_\ell \in U_{\ell,\bar{\mu}[\ell]}(x)} H\big(x, \bar{\mu}_1[\ell](x), \ldots, \bar{\mu}_{\ell-1}[\ell](x), u_\ell, \bar{\mu}_{\ell+1}[\ell](x), \ldots, \bar{\mu}_m[\ell](x), \overline{J}\big), \qquad \ell = 1, \ldots, m, \ x \in X. \tag{3.8}$$

Eq. (3.7) and the contraction property of $T_{\bar{\mu}[\ell]}$ imply that $\overline{J}$ is equal to the cost functions $J_{\bar{\mu}[\ell]}$ of all of the m policies $\bar{\mu}[\ell]$, $\ell = 1, \ldots, m$. In view of the uniqueness Assumption 3.1, this implies that all the policies $\bar{\mu}[\ell]$, $\ell = 1, \ldots, m$, are equal to some policy $\bar{\mu}$, which has cost function $\overline{J}$, and in view of Eq. (3.8), satisfies

$$\overline{J}(x) = \min_{u_\ell \in U_{\ell,\bar{\mu}}(x)} H\big(x, \bar{\mu}_1(x), \ldots, \bar{\mu}_{\ell-1}(x), u_\ell, \bar{\mu}_{\ell+1}(x), \ldots, \bar{\mu}_m(x), \overline{J}\big), \qquad \ell = 1, \ldots, m. \tag{3.9}$$

It follows that $\bar{\mu}$ is agent-by-agent optimal.

Finally, the preceding argument shows that $\overline{J}$ is the cost function of every policy that is repeated infinitely often. Thus the uniqueness Assumption 3.1 implies that $\bar{\mu}$ is the only policy that is repeated infinitely often. Since there are finitely many policies, it follows that $\mu^k = \bar{\mu}$ for all k after some index. Hence from the definition of the algorithm, the sequence $\{J^k\}$ satisfies $J^{k+1} = T_{\bar{\mu}} J^k$ for all k after some index, which in view of the contraction Assumption 1.2, implies Eq. (3.4). □

Note that the preceding proposition does not guarantee convergence to the optimal policy (which is unique by Assumption 3.1). In particular, if our algorithm is started at a pair $(J_{\bar{\mu}}, \bar{\mu})$, where $\bar{\mu}$ is an agent-by-agent optimal policy, it will not move from $\bar{\mu}$ [in fact it can be shown that this will happen even if the algorithm is started at a pair $(J^0, \bar{\mu})$, where J^0 is sufficiently close to $J_{\bar{\mu}}$]. Thus every agent-by-agent optimal policy behaves like a "local minimum", with its own "region of attraction", and is a potential convergence limit of our algorithm. The limit will depend on the starting pair, as well as the order in which the agents select their components. The algorithm guarantees convergence to the optimal policy only under additional assumptions that guarantee that there are no additional agent-by-agent optimal policies. We postpone a discussion of this issue for Section 5.

Ensuring convergence to an optimal policy with randomization schemes

Another possibility to enhance the convergence properties of the algorithm, and ensure convergence to an optimal policy, is to enlarge the constraint sets

$$U_{\ell,(\hat{\mu}_1,\ldots,\hat{\mu}_{\ell-1},\mu_\ell,\ldots,\mu_m)}(x)$$

in Eq. (2.4), to allow minimization over subsets of multiple control components. These subsets may be selected with some form of randomization: at some iterations minimize over a single control component as in iteration (2.4)–(2.5), while at some other randomly chosen iterations minimize over multiple or even all control components. Schemes of this type have been considered for the purpose of enhancing the convergence properties of asynchronous PI; see [3], Section 2.5.3. Randomization over sets of multiple control components can also be used in the context of the optimistic agent-by-agent PI methods of the next section, and they can similarly enhance their convergence properties.

We will not consider randomized control component selection schemes in this paper. Their analysis is similar to the one of [3], Section 2.5.3, their implementation is likely problem-dependent, and their practical performance is an interesting subject for further research. Their principal drawback is that simultaneous minimization over multiple control components can be very costly (depending on the number of components involved), even if it used in only a small proportion of the total number of iterations.

4. Agent-by-agent optimistic policy iteration

Let us now consider an optimistic PI variant where we introduce an infinite subset $\mathcal{K} \subset \{0, 1, \ldots\}$ of the iterations, and the complementary subset of iterations $k \notin \mathcal{K}$. For the latter subset, we use the (less expensive) standard policy evaluation update $J^{k+1} = T_{\mu^k} J^k$, and no policy update:

$$(J^{k+1}, \mu^{k+1}) \in \widetilde{\mathcal{P}}(J^k, \mu^k), \qquad \forall\ k \in \mathcal{K}, \tag{4.1}$$

$$J^{k+1} = T_{\mu^k} J^k, \qquad \mu^{k+1} = \mu^k, \qquad \forall\ k \notin \mathcal{K}. \tag{4.2}$$

We have the following convergence result:

Proposition 4.1 (*Optimistic PI Convergence to an Agent-by-Agent Optimal Policy*)**.** *Let the assumptions of Proposition 3.1 hold, and let $\{J^k, \mu^k\}$ be a sequence generated by the optimistic agent-by-agent PI algorithm (4.1)–(4.2). Then there is an index $\overline{k}$ such that for all $k \geq \overline{k}$, we will have $\mu^k = \bar{\mu}$, where $\bar{\mu}$ is an agent-by-agent optimal policy, while the sequence $\{J^k\}$ will converge to $J_{\bar{\mu}}$.*

Proof. The proof is essentially identical to the one of Proposition 3.1. In particular, the definition of the optimistic PI algorithm allows the proof of the critical relation (3.5) to go through. □

The algorithm admits also a distributed implementation, whereby the iteration (4.1)–(4.2) is executed at the subset of times $k \in \mathcal{K}$ only for a subset X_k of the states, while for the remaining states $x \notin X_k$ the values of $J^{k+1}(x)$ and $\mu^{k+1}(x)$ remain unchanged:

$$J^{k+1}(x) = J^k(x), \qquad \mu^{k+1}(x) = \mu^k(x), \qquad \forall\ x \notin X_k,\ k \in \mathcal{K}. \tag{4.3}$$

In addition to the set $\mathcal{K}$ being infinite, there is a requirement here is that each state x belongs infinitely often to some subset X_k, so that there are infinitely many policy improvements at every state. Algorithms of this type have been proposed in the book [4], Section 2.2.3, and in [5]. The convergence proof of Proposition 3.1 still goes through; see also the proof of Prop. 2.5 of [4]. Note, however, that for this type of algorithm to be provably convergent, (J^0, μ^0) must satisfy the condition $T_{\mu^0} J^0 \leq J^0$ [cf. Eq. (3.2)] even for discounted MDP, as demonstrated with counterexamples by Williams and Baird [22] (see also [23]).

In a more complex version of the algorithm, the information on the cost function iterates at each iteration is allowed to be out-of-date, while modifications are introduced to eliminate the need for the initial condition assumption of Eq. (3.2). Distributed asynchronous PI algorithms of this type have been proposed and analyzed in the paper by Bertsekas and Yu [24] [see also [25,26], and the books [5] (Section 2.6), and [3] (Section 2.6)]. See also the randomized optimistic PI algorithms of [3] (Section 2.5.3). Multiagent versions of such algorithms are a subject for further research.

5. Conditions for obtaining an optimal policy

We proved earlier that our multiagent VI algorithm will find an agent-by-agent optimal policy under our assumptions of Proposition 3.1, but this policy need not be optimal. We will now discuss approaches that can be used to show that the policy obtained is optimal, under the same or alternative assumptions. One possibility is to impose conditions under which every agent-by-agent optimal policy is optimal. To this end we introduce the following definition.

Definition 5.1 (*Component-by-Component Minimum*). For a state x and a function $J \in \mathcal{R}(X)$ we say that a control $\overline{u} = (\overline{u}_1, \ldots, \overline{u}_m) \in U(x)$ is a *component-by-component minimum of H at (x, J)* if

$$\overline{u}_\ell \in \arg\min_{u_\ell \in \overline{U}_{\ell,\overline{u}}(x)} H(x, \overline{u}_1, \ldots, \overline{u}_{\ell-1}, u_\ell, \overline{u}_{\ell+1}, \ldots, \overline{u}_m, J), \qquad \ell = 1, \ldots, m, \tag{5.1}$$

where the sets $\overline{U}_{\ell,\overline{u}}(x)$ are defined by

$$\overline{U}_{\ell,\overline{u}}(x) = \big\{u_\ell \mid (\overline{u}_1, \ldots, \overline{u}_{\ell-1}, u_\ell, \overline{u}_{\ell+1}, \ldots, \overline{u}_m) \in U(x)\big\}, \qquad \ell = 1, \ldots, m.$$

Note that from the definition of agent-by-agent optimality, we have that $\bar{\mu}$ is agent-by-agent optimal if for every $x \in X$, the control $\bar{\mu}(x)$ is a component-by-component minimum of H at $(x, J_{\bar{\mu}})$. We have the following proposition.

Proposition 5.1 (*Agent-by-Agent Optimality Criterion*). *Assume that for every state $x \in X$ and policy $\bar{\mu}$ such that $\bar{\mu}(x)$ is a component-by-component minimum of H at $(x, J_{\bar{\mu}})$, the control $\bar{\mu}(x)$ minimizes $H(x, u, J_{\bar{\mu}})$ over $u \in U(x)$. Then every agent-by-agent optimal policy is optimal.*

Proof. Let $\bar{\mu}$ be agent-by-agent optimal. Then from the definition of agent-by-agent optimality, we have that for all $x \in X$, $\bar{\mu}(x)$ is a component-by-component minimum of H at $(x, J_{\bar{\mu}})$. By our assumption, this implies that for all $x \in X$, $\bar{\mu}(x)$ minimizes $H(x, u, J_{\bar{\mu}})$ over $u \in U(x)$, or $T_{\bar{\mu}} J_{\bar{\mu}} = T J_{\bar{\mu}}$. From general properties of contractive abstract DP models (cf. [3], Chapter 2), we also have $T_{\bar{\mu}} J_{\bar{\mu}} = J_{\bar{\mu}}$. Hence $T_{\bar{\mu}} J_{\bar{\mu}} = T J_{\bar{\mu}}$, which implies that $J_{\bar{\mu}} = J^*$ (cf. [3], Chapter 2), so $\bar{\mu}$ is optimal. □

In view of Proposition 5.1, an important issue is to delineate sufficient conditions that guarantee that component-by-component minima of H at (x, J_μ) minimize $H(x, u, J_\mu)$ over $u \in U(x)$. Somewhat similar questions have been addressed in two related contexts:

(a) Team theory in connection with the notion of person-by-person optimality mentioned earlier.
(b) The theory of convergence of coordinate descent methods in nonlinear optimization.

In the theory of teams and other related decentralized control problem formulations, the most prominent analytical issues arise when the team members select control components based on different information. By contrast in our framework the agents choose actions based on shared information, namely the current state x_k of the system. Because of this fundamental structural assumption, DP algorithms such as VI and PI apply to our problem, but do not apply to team problems with nonclassical information pattens. These problems are generally far more complicated than the ones considered here, as illustrated for linear systems and quadratic cost by the famous counterexample of [27].

In the theory of coordinate descent methods, the result most related to our context is that if a function $F(y_1, \ldots, y_m)$ of m vectors $y_1, \ldots, y_m$ is strictly convex and differentiable over the Cartesian product $Y_1 \times \cdots \times Y_m$ of closed convex sets $Y_1, \ldots, Y_m$, then a vector $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m)$ is a global minimum of F over $Y_1 \times \cdots \times Y_m$ if and only if it has the component-by-component minimization property

$$\bar{y}_\ell \in \arg\min_{y_\ell \in Y_\ell} F(\bar{y}_1, \ldots, \bar{y}_{\ell-1}, y_\ell, \bar{y}_{\ell+1}, \ldots, \bar{y}_m), \qquad \text{for all } \ell = 1, \ldots, m. \tag{5.2}$$

Thus when F is strictly convex and differentiable, the block coordinate descent method cannot get trapped into a solution that is a component-by-component minimum but is not a global minimum [this is not true, however, if F is strictly convex but nondifferentiable, since the condition (5.2) may hold at vectors $\bar{y}$ that are not global minima, and at which F is nondifferentiable]. Some related results are known for the case where the sets Y_ℓ are discrete, under assumptions that can be viewed as discrete space substitutes for strict convexity; see e.g., de Waal and van Schuppen [15], and Bauso and Pesenti [28,29].

While the coordinate descent and the team theory results provide some analytical guidance, they do not apply directly to the DP context of this paper. The reason is that the mapping H involves the functions J_μ, whose properties have to be verified through analysis. We leave this line of investigation as a subject for further research, and we outline another analytical approach, which assumes continuous state and control spaces X and U, and is based on strict convexity and differentiability assumptions.

Continuous spaces, strict convexity, and differentiability

Let us remove the assumption that the state and control spaces X and U are finite, while continuing to assume that the control has m components, $u = (u_1, \ldots, u_m)$ that are constrained by $u \in U(x)$ for all $x \in X$. We continue to adopt the monotonicity and contraction Assumptions 1.1 and 1.2, with the modification that $\mathcal{R}(X)$ is replaced by the space $\mathcal{B}(X)$ of bounded functions over X, with respect to a weighted sup-norm. Moreover, we assume that the various minima over control components in the definition of the algorithms are attained. Models of this type have been analyzed extensively in the monograph [3] (Chapter 2), to which we refer for a detailed discussion. The definitions of agent-by-agent optimality and component-by-component minimum carry over without change to the continuous spaces setting, and so does the associated agent-by-agent optimality criterion (cf. Proposition 5.1). Furthermore, the key inequality (3.4) for the proof of the convergence result of Proposition 3.1 goes through, under the condition $T_{\mu^0} J^0 \leq J^0$ [cf. Eq. (3.2)]. As a result, the proof of monotonic decrease of the sequence $\{J^k\}$ to some function $\overline{J}$ goes through as well.

In conclusion, without assuming finiteness of the state and control spaces X and U, our algorithm, under the monotonicity and contraction Assumptions 1.1 and 1.2, and the condition $T_{\mu^0} J^0 \leq J^0$ [cf. Eq. (3.2)], converges monotonically to some $\overline{J}$, which can be seen to be pointwise bounded below by the optimal cost function J^*, which belongs to $\mathcal{B}(X)$, so that $\overline{J} \in \mathcal{B}(X)$. Further conditions, involving strict convexity and differentiability, need to be imposed to guarantee that $\overline{J} = J^*$, that J^* is convex and differentiable, and that an optimal policy can be obtained. A stochastic optimal control model, involving a linear system, a convex cost per stage, and convex state and control constraints, was formulated and analyzed in 1973 by the author [30], and is well suited for this purpose. We leave further analysis along this line as a subject for further research.

6. Concluding remarks

We have shown that in the context of multiagent problems, agent-by-agent versions of the VI algorithm and related optimistic PI algorithms have greatly reduced computational requirements, while still maintaining a meaningful convergence property. While these algorithms may terminate with a suboptimal policy that is agent-by-agent optimal, they can be dramatically more efficient than the standard VI and optimistic PI algorithms, which may be computationally intractable even for a moderate number of agents.

Several unresolved questions remain regarding algorithmic variations and conditions that guarantee that our algorithms obtain an optimal policy rather than one that is agent-by-agent optimal. Approximate versions of our algorithms of the type used in neuro-dynamic programming/reinforcement learning are also of interest, and are a subject for further investigation. Moreover, the basic idea of our approach, simplifying the minimization defining the VI operator while maintaining some form of convergence guarantee, can be extended in other directions to exploit special problem structures.

Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.rico.2020.100003.

References

[1] Bertsekas DP. Multiagent rollout algorithms and reinforcement learning. 2020, arXiv preprint, arXiv:2002.07407.
[2] Bertsekas DP. Multiagent reinforcement learning: Rollout and policy iteration. IEEE/CAA J Autom Sin 2020 [in press].
[3] Bertsekas DP. Abstract dynamic programming. Belmont, MA: Athena Scientific; 2018, On-line at http://web.mit.edu/dimitrib/www/RLbook.html.
[4] Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. Belmont, MA: Athena Scientific; 1996.
[5] Bertsekas DP. Dynamic programming and optimal control, vol. II. 4th ed. Belmont, MA: Athena Scientific; 2012.
[6] Bertsekas DP. Reinforcement learning and optimal control. Belmont, MA: Athena Scientific; 2019.
[7] Marschak J. Elements for a theory of teams. Manage Sci 1975;1:127–37.
[8] Radner R. Team decision problems. Ann Math Stat 1962;33:857–81.
[9] Witsenhausen H. On information structures, feedback, causality. SIAM J Control 1971;9:149–60.
[10] Witsenhausen H. Separation of estimation and control for discrete time systems. Proc IEEE 1971;59:1557–66.
[11] Witsenhausen H. Equivalent stochastic control problems. Math Control Signals Systems 1988;1:3–11.
[12] Ho YC. Team decision theory and information structures. Proc IEEE 1980;68:644–54.
[13] Krainak JC, Speyer J, Marcus S. Static team problems - part I: Sufficient conditions and the exponential cost criterion. IEEE Trans Automat Control 1982;27:839–48.
[14] Krainak JC, Speyer J, Marcus S. Static team problems - part II: Affine control laws, projections, algorithms, and the LEGT problem. IEEE Trans Automat Control 1982;27:848–59.
[15] de Waal PR, van Schuppen JH. A class of team problems with discrete action spaces: Optimality conditions based on multimodularity. SIAM J Control Optim 2000;38:875–92.
[16] Nayyar A, Mahajan A, Teneketzis D. Decentralized stochastic control with partial history sharing: A common information approach. IEEE Trans Automat Control 2013;58:1644–58.
[17] Nayyar A, Teneketzis D. Common knowledge and sequential team problems. IEEE Trans Automat Control 2019;64:5108–15.
[18] Li Y, Tang Y, Zhang R, Li N. Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach. 2019, arXiv preprint arXiv:1912.09135.
[19] Gupta A. Existence of team-optimal solutions in static teams with common information: A topology of information approach. SIAM J Control Optim 2020;58:998–1021.
[20] Zoppoli R, Sanguineti M, Gnecco G, Parisini T. Neural approximations for optimal control and decision. Springer; 2020.
[21] Bertsekas DP. Nonlinear programming. 3rd ed. Belmont, MA: Athena Scientific; 2016.

[22] Williams RJ, Baird LC. Analysis of some incremental variants of policy iteration: First steps toward understanding actor-critic learning systems. Report NU-CCS-93-11, Boston, MA: College of Computer Science, Northeastern Univ.; 1993.
[23] Bertsekas DP. Williams-baird counterexample for Q-factor asynchronous policy iteration. 2010, http://web.mit.edu/dimitrib/www/Williams-BairdCounterexample.pdf.
[24] Bertsekas DP, Yu H. Asynchronous distributed policy iteration in dynamic programming. In: Proc. of allerton conf. on communication, control and computing. Ill: Allerton Park; 2010, p. 1368–74.
[25] Bertsekas DP, Yu H. Q-learning and enhanced policy iteration in discounted dynamic programming. Math. OR 2012;37:66–94.
[26] Yu H, Bertsekas DP. Q-learning and policy iteration algorithms for stochastic shortest path problems. Ann Oper Res 2013;208:95–132.
[27] Witsenhausen H. A counterexample in stochastic optimum control. SIAM J Control 1968;6:131–47.
[28] Bauso D, Pesenti R. Generalized person-by-person optimization in team problems with binary decisions. In: Proc. 2008 American control conference. 2008. p. 717-22.
[29] Bauso D, Pesenti R. Team theory and person-by-person optimization with binary decisions. SIAM J Control Optim 2012;50:3011–28.
[30] Bertsekas DP. Linear convex stochastic control problems over an infinite horizon. IEEE Trans Aut Control 1973;AC-18:314–5.