Automatica 61 (2015) 0–13

Contents lists available at ScienceDirect

Automatica

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

An algorithmic approach to identify irrelevant information in sequential teams[☆]

Aditya Mahajan[a], Sekhar Tatikonda[b]

[a] Department of Electrical and Computer Engineering, McGill University, Montreal, Canada
[b] Department of Electrical Engineering, Yale University, New Haven, USA

ARTICLE INFO

Article history:
Received 8 August 2013
Received in revised form
16 June 2015
Accepted 28 July 2015

Keywords:
Sequential teams
Decentralized stochastic control
Markov decision theory
Graphical models
Bayesian networks

ABSTRACT

An algorithmic framework that identifies irrelevant data (i.e., data that may be ignored without any loss of optimality) at agents of a sequential team is presented. This framework relies on capturing the properties of a sequential team that do not depend on the specifics of state spaces, the probability law, the system dynamics, or the cost functions. To capture these properties the notion of a team form is developed. A team form is then modeled as a directed acyclic graph and irrelevant data is identified using D-separation properties of specific subsets of nodes in the graph. This framework provides an algorithmic procedure for identifying and ignoring irrelevant data at agents, and thereby simplifying the form of control laws that need to be implemented.

1. Introduction

Teams refer to multi-agent stochastic control systems in which all agents have a common objective. Teams arise in many modern technologies including networked control systems, communication networks, sensor and surveillance networks, environmental remote sensing, and smart grids. Dynamic programming, which is the main solution concept for optimal design of centralized stochastic control, only works for specific sub-classes of team problems (Nayyar, Mahajan, & Teneketzis, 2013). To apply the dynamic programming principle to general team problems, one needs to identify *the structure of optimal control laws*. Such structural results are of two type: (i) remove irrelevant information at the controller; (ii) identify a sufficient statistic of the data available at the controller. In this paper, we present an algorithmic approach to identify the first type of structural result. As an example of such structural results, consider the problem of real-time communication studied in Witsenhausen (1979).

Example 1. Consider a real-time communication system consisting of a source, an encoder, and a decoder. The source is a first-order Markov process $\{S_t\}_{t=1}^{\infty}$. The encoder observes the source output and generates quantized symbols Q_t, causally and in real-time, as follows

$$Q_t = e_t(S_{1:t}, Q_{1:t-1})$$

where $S_{1:t}$ is a short hand notation for $(S_1, \dots, S_t)$ and $Q_{1:t-1}$ has a similar interpretation. The decoder is a finite state machine. M_t denotes the state of the machine at time t. The decoder generates an estimate $\hat{S}_t$ of the source as follows

$$\hat{S}_t = d_t(Q_t, M_{t-1})$$

and updates the contents of its memory as follows

$$M_t = g_t(Q_t, M_{t-1}).$$

At each time a distortion $c_t(S_t, \hat{S}_t)$ is incurred. The objective is to choose an encoding policy $\mathbf{e} := (e_1, e_2, \dots, e_T)$, a decoding policy $\mathbf{d} := (d_1, d_2, \dots, d_T)$, and a memory update policy $\mathbf{g} := (g_1, g_2, \dots, g_T)$ to minimize

$$\mathbb{E}_{(\mathbf{e},\mathbf{d},\mathbf{g})}\Big[\sum_{t=1}^{T} c_t(S_t, \hat{S}_t)\Big].$$

[☆] A. Mahajan was supported by the Natural Science and Engineering Research Council of Canada (NSERC) Grant NSERC-RGPIN 402753-11 and S. Tatikonda was partially supported by National Science Foundation (NSF) grant CCF-1017744. The material in this paper was partially presented at the 2009 ConCom (Control over communication channels) workshop, June 27, 2009, Seoul, South Korea (Mahajan & Tatikonda, 2009a) and at the 47th Allerton conference on communication, control, and computing, September 30–October 2, 2009, Monticello, IL, USA (Mahajan & Tatikonda, 2009b). This paper was recommended for publication in revised form by Associate Editor Valery Ugrinovskii under the direction of Editor Ian R. Petersen.

E-mail addresses: aditya.mahajan@mcgill.ca (A. Mahajan), sekhar.tatikonda@yale.edu (S. Tatikonda).

http://dx.doi.org/10.1016/j.automatica.2015.08.002
0005-1098/© 2015 Elsevier Ltd. All rights reserved.

The following two structural results hold for Example 1.

(1) For any decoding and memory update strategies $(\mathbf{d}, \mathbf{g})$, there is no loss of optimality in restricting attention to an encoding strategy of the form

$$Q_t = e_t(S_t, M_{t-1}).$$

(2) When $M_{t-1} = Q_{1:t-1}$, there is no loss of optimality in restricting attention to encoding and decoding strategies of the form

$$Q_t = e_t(S_t, \Pi_t) \quad \text{and} \quad \hat{S}_t = d_t(Q_t, \Pi_t)$$

where $\Pi_t(s) = \mathbb{P}(S_t = s \mid Q_{1:t-1})$.

The first structural result shows that $S_{1:t-1}$ and $Q_{1:t-1}$ are *irrelevant* at the encoder. The second structural result shows that Π_t is a sufficient statistic for $Q_{1:t-1}$. The first structural result was proved in Witsenhausen (1979) and the second in Walrand and Varaiya (1983). Based on the second structural result, a dynamic programming decomposition was presented in Walrand and Varaiya (1983).

In this paper, we develop an algorithmic framework to identify irrelevant data at agents in a team. Removing such irrelevant data is usually the first step in deriving a dynamic programming decomposition for teams. For example, in the above example, the derivation of the second structural result (and therefore of the dynamic program) relies on the first structural result.

Structural results that remove irrelevant data are robust to various modeling assumptions: the specifics of the state spaces, the underlying probability measure, and the specifics of the plant dynamics and the cost functions. All that matters is the *form* of the system. We model dynamical systems using directed acyclic graph in such a manner that captures the *form* of a team. Removing irrelevant data is equivalent to removing edges from the corresponding directed acyclic graph. To identify the irrelevant data, we use graphical modeling algorithms to iteratively apply Blackwell's principle of irrelevant information (Blackwell, 1964), which we state below for completeness.

Theorem 1 (*Blackwell's Principle of Irrelevant Information*)**.** *For any Borel spaces $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{U}$, let P be a probability measure on $\mathbb{X} \times \mathbb{Y}$ and $c : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ be a bounded Borel-measurable function. Then for any Borel-measurable function $g : \mathbb{X} \times \mathbb{Y} \to \mathbb{U}$, there exists another Borel measurable function $h : \mathbb{X} \to \mathbb{U}$ such that*

$$\mathbb{E}[c(x, h(x))] \le \mathbb{E}[c(x, g(x, y))]$$

where the expectation is with respect to P.

A consequence of Theorem 1 is the following. Consider the optimization problem of choosing a control law $g : \mathbb{X} \times \mathbb{Y} \to \mathbb{U}$ to minimize $\mathbb{E}[c(x, g(x, y))]$. Then, there is no loss of optimality in restricting attention to control laws of the form $h : \mathbb{X} \to \mathbb{U}$. Equivalently, the observation y is irrelevant for optimal control. In this paper, we present algorithms that recursively apply Blackwell's principle at each agents and groups of agents to identify irrelevant data in teams.

1.1. Literature overview

Team problems were introduced in the economics literature in the 1950s (Marschak & Radner, 1972; Radner, 1962) and have been extensively analyzed in the control literature since the 1970s (Ho, 1980; Sandell, Varaiya, Athans, & Safonov, 1978; Witsenhausen, 1971). Motivated by applications in networked control systems, there has been tremendous activity in the study of team problems in the last decade. We refer the reader to Mahajan, Martins, Rotkowitz, and Yüksel (2012) and references therein for a detailed literature overview.

Broadly speaking, team problems are modeled either in state space using information structures or in input–output formulation using sparsity constraints. We follow the former modeling paradigm in this paper. Such models are analyzed either for the LQG setup (linear dynamics, quadratic cost, and Gaussian disturbances) or general (non-linear) Markovian setup. In this paper, we follow the latter setup and develop an algorithmic procedure to identify and remove irrelevant data at each agent.

We model teams using a directed acyclic graph (DAG) and use algorithms from graphical models to remove edges that correspond to irrelevant data. A DAG is a natural structure to model the causality and partial order relationship between the system variables of a sequential team. Other researchers have also used DAGs to model sequential teams (Gattami, 2007; Ho & Chu, 1972; Witsenhausen, 1971; Yoshikawa, 1978) but, to the best of our knowledge, the idea of using graphical modeling algorithms on the DAG representation to identify and remove redundant information has not been used before.

1.2. Contributions

Our main contribution is to present a graphical model for sequential team. This model captures the information structure of the system and the conditional independence relations between all system variables.

Using this graphical mode, we develop graphical modeling algorithms that identify irrelevant data at each agent. An agent can ignore this data without any loss of optimality. Two such algorithms are presented. The first algorithm sequentially identifies irrelevant data at each agent in the system. Preliminary versions of this algorithm were presented in Mahajan & Tatikonda, 2009a,b. The second algorithm sequentially identifies irrelevant data at all possible subsets of agents in the system. These algorithms do not depend on the type of system dynamics or the cost function.

The rest of the paper is organized as follows. In Section 2 we define team form and team type and formulate the problem of simplification of a team form. In Section 3 we present background material on graphical models and in Section 4 we describe how to represent a team form using a DAG (directed acyclic graph). Simplification of a team form may be viewed as removing edges from this DAG. Algorithms that perform this simplification are presented in Section 5 (for a single agent) and Section 6 (for a group of agents). Examples illustrating this approach are presented in Section 7 and we conclude in Section 8.

1.3. Notation

We use the following notation in the paper.

- For a set A, $|A|$ denotes the cardinality of A.
- For two sets A and B, $A \times B$ denotes their Cartesian product.
- For two measurable spaces $(\mathbb{X}, \mathscr{F})$ and $(\mathbb{Y}, \mathscr{G})$, $\mathscr{F} \otimes \mathscr{G}$ denotes the product σ-field on $\mathbb{X} \times \mathbb{Y}$.
- For two probability measures μ on $(\mathbb{X}, \mathscr{F})$ and ν on $(\mathbb{Y}, \mathscr{G})$, $\mu \otimes \nu$ denotes the product probability measure on $\mathscr{F} \otimes \mathscr{G}$.
- $X_{1:t}$ is a short hand for the sequence $(X_1, X_2, \dots, X_t)$.
- For a set N and a sequence of random variables $\{X_n\}$, X_N is a short-hand for $(X_n : n \in N)$.
- For a set N and a sequence of state spaces $\{\mathbb{X}_n\}$, $\mathbb{X}_N$ is a short-hand for $\prod_{n \in N} \mathbb{X}_n$.
- For a set N and a sequence of σ-fields $\{\mathscr{F}_n\}$, $\mathscr{F}_N$ is a short-hand for $\bigotimes_{n \in N} \mathscr{F}_n$.

2. Modeling sequential team using team form and team type

A sequential team is a decentralized control system consisting of multiple agents (also called controllers or decision makers), indexed by a set A. Agent α, $\alpha \in A$, chooses a *control action* U_α. Nature chooses $|B|$ *primitive random variables* $\{W_\beta : \beta \in B\}$. These variables are chosen independently according to a product probability measure Q. The *system variables* $\{S_m : m \in M\}$ are generated when the system evolves with time. All these variables are collectively indexed by $N = A \cup B \cup M$ and denoted by a generic term X_n. Variable X_n, $n \in N$, takes values in a measurable space $(\mathbb{X}_n, \mathscr{F}_n)$.

Agent $\alpha, \alpha \in A$, observes some data $\{X_n : n \in I_\alpha\}$; the set $I_\alpha \subseteq N$ is called the *information set* of agent α. Agent α chooses action X_α (also denoted by U_α), according to a *control law* $g_\alpha : \left(\mathbb{X}_{I_\alpha}, \mathscr{F}_{I_\alpha}\right) \to (\mathbb{X}_\alpha, \mathscr{F}_\alpha)$ as $X_\alpha = g_\alpha(X_{I_\alpha})$. The collection $\mathbf{g} := (g_\alpha : \alpha \in A)$ is called a *control policy*.

Each system variable X_m (also denoted by S_m), $m \in M$, is also associated with an information set I_m and a *dynamics function* $f_m : \left(\mathbb{X}_{I_m}, \mathscr{F}_{I_m}\right) \to (\mathbb{X}_m, \mathscr{F}_m)$. The variable X_m is generated as $X_m = f_m(X_{I_m})$. The collection $(f_m : m \in M)$ is called the *system dynamics*.

The system is *sequential* (Witsenhausen, 1975). This means that a bijection $\varphi : N \to \{1, 2, \dots, |N|\}$ exists such that for any $n \in \varphi(A \cup M)$, $I_{\varphi^{-1}(n)} \subseteq \varphi^{-1}(\{1, 2, \dots, n-1\})$. Consequently, we can impose a total order on all the system variables such that the total order is consistent with the causal relationship between the variables and, at the same time, does not depend on the realization of the primitive random variables or the choice of control policy. As shown in Witsenhausen (1975), a sequential system has the following equivalent representation. Define a binary operator $\leftarrow$ on $A \cup M$ such that for $n, m \in A \cup M$, $n \leftarrow m$ iff $n \in I_m$. Then a system is sequential if and only if the transitive closure of $\leftarrow$ is a partial order. We will use this latter characterization in this paper.

In a sequential system, any choice $\mathbf{g}$ of a control policy induces a probability measure $P_{\mathbf{g}}$ on all system variables $\{X_n : n \in N\}$ which is given by

$$P_{\mathbf{g}}(X_N) = \bigotimes_{m \in M} \delta(X_m = f_m(X_{I_m})) \otimes \bigotimes_{\alpha \in A} \delta(X_\alpha = g_\alpha(X_{I_\alpha})) \otimes \bigotimes_{\beta \in B} Q(X_\beta) \tag{1}$$

where $\delta(\cdot)$ is a Dirac measure. The performance of a control policy is quantified by expectation (with respect to $P_{\mathbf{g}}$) of a cost function $\sum_{k \in K} c_k(X_{D_k})$ where K is a finite set, the sets D_k, $k \in K$, are subsets of N and $c_k : \left(\mathbb{X}_{D_k}, \mathscr{F}_{D_k}\right) \to (\mathbb{R}, \mathscr{B})$, where $\mathscr{B}$ is the Borel σ-algebra over reals.

Formally, a sequential team is specified by two parts: the team form and the team type.

Definition 1 (*Team Form*). A *team form* is specified by three components:

(1) *Variable Structure* (N, A, B, M) where:
- N is a finite set that indexes all the variables of the system
- A, B, and M are disjoint subsets of N such that $A \cup B \cup M = N$.
- The set A indexes the agents and their control actions.
- The set B indexes the primitive random variables.
- The set M indexes the system variables.

(2) *Information Structure* $\{I_n : n \in A \cup M\}$:
- For any $n \in A \cup M$, $I_n \subseteq N$.
- For $n \in A$, the set I_n denotes the information set of agent n (i.e., the set of all variables observed by agent n).
- For $n \in M$, the set I_n denotes the information set of variable X_n (i.e., the set of variables coupled to X_n through the dynamics).
- The collection $\{I_n : n \in A \cup M\}$ satisfies the following property. Define a binary relation $\leftarrow$ on $A \cup M$ such that for $n, m \in A \cup M$, $n \leftarrow m$ if and only if $n \in I_m$. The transitive closure of $\leftarrow$ is a partial order on $A \cup M$.

(3) *Cost Structure* $(K, \{D_k : k \in K\})$:
- The set K indexes the coupled cost terms
- The sets $D_k \subset N$, $k \in K$ denote the variables coupled by the kth cost term.

Definition 2 (*Team Type*). A *team type* is specified by four components

(1) *Measurable Spaces* $\{(\mathbb{X}_n, \mathscr{F}_n) : n \in N\}$:
- Variable X_n, $n \in N$ takes value in measurable space $(\mathbb{X}_n, \mathscr{F}_n)$.

(2) *Probability Measure* Q:
- Q is a product measure on the primitive random variables $\{X_\beta : \beta \in B\}$.

(3) *System dynamics* $\{f_m : m \in M\}$:
- $f_m : \left(\mathbb{X}_{I_m}, \mathscr{F}_{I_m}\right) \to (\mathbb{X}_m, \mathscr{F}_m)$ is a measurable function such that $X_m = f_m(X_{I_m})$.

(4) *Cost functions* $\{c_k : k \in K\}$:
- $c_k : \left(\mathbb{X}_{D_k}, \mathscr{F}_{D_k}\right) \to (\mathbb{R}, \mathscr{B})$ denotes the kth cost function. The total cost incurred in the system is $\sum_{k \in K} c_k(X_{D_k})$.

Definition 3 (*Control Policy*). A control policy for a sequential team is a collection $\mathbf{g} := \{g_\alpha : \alpha \in A\}$ such that

$$g_\alpha : \left(\mathbb{X}_{I_\alpha}, \mathscr{F}_{I_\alpha}\right) \to (\mathbb{X}_\alpha, \mathscr{F}_\alpha).$$

Given a control policy, the control variables are generated as $X_\alpha = g_\alpha(X_{I_\alpha})$.

We are interested in the following stochastic control problem.

Problem 1. Given a team form and a team type, choose a control policy $\mathbf{g}$ to minimize $\mathbb{E}_{\mathbf{g}}\left[\sum_{k \in K} c_k(X_{D_k})\right]$ where the expectation is with respect to the induced probability measure $P_{\mathbf{g}}$. The corresponding minimum cost is called the *value* of the team.

Remark 1. (1) Instead of using primitive random variables, an equivalent way to describe system dynamics is using a Markov kernel on the next state conditioned on the current state and the control action.

(2) There is no loss of generality in assuming that the primitive random variables are independent. Correlated primitive random variables may be viewed as system variables generated from independent primitive random variables.

(3) Our definition of a control law does not rule out the possibility of randomized control laws. A randomized control law is in fact a deterministic control law where the agent observes an additional "randomizing" primitive random variable that is used to pick the control action.

(4) The model described above is similar to the model considered in Witsenhausen (1988), which, in turn, was shown to be equivalent to the intrinsic model (Witsenhausen, 1971, 1975) when specialized to sequential teams. By a similar argument, the above model is equivalent to the intrinsic model.

(5) Throughout this paper, we will assume that for all $n \in N$, the σ-algebras $\mathscr{F}_n$ are countably generated and contain all singletons of $\mathbb{X}_n$. Borel σ-algebras satisfy these properties.

(6) Since N is finite and for all $k \in K$, $D_k \subseteq N$, we are implicitly assuming that K is finite as well. Thus, the discussion in this paper is restricted to finite horizon systems.

As an example, the team form corresponding to the real-time communication system of Example 1 is given as follows. First note that a first-order Markov process $\{S_t\}_{t=1}^T$ may be represented as $S_t = \varphi_t(S_{t-1}, W_{t-1})$ using the inverse transform method or Smirnov transform (Grimmett & Stirzaker, 2001, Sec 4.11) such that $\{W_t\}_{t=1}^T$ is an independent process that is also independent of S_1. With this representation, the team form of Example 1 is given by:

(1) *Variable structure*
- Control variables: $X_A = \{Q_{1:T}, M_{1:T-1}, \hat{S}_{1:T}\}$.
- Primitive variables: $X_B = \{S_1, M_0, W_{1:T}\}$.
- System variables: $X_M = \{S_{2:T}\}$.
- All variables: $X_N = X_A \cup X_B \cup X_M$.

(2) *Information structure*
- For $X_n = Q_t \in X_A$, the information set I_n is $\{S_{1:t}, Q_{1:t-1}\}$.
- For $X_n = M_t \in X_A$, the information set I_n is $\{Q_t, M_{t-1}\}$.
- For $X_n = \hat{S}_t \in X_A$, the information set I_n is $\{S_{t-1}, W_{t-1}\}$.

(3) *Cost structure*
- Set of coupled costs $K = \{1, \dots, T\}$.
- For $t = 1, 2, \dots, T, D_t = \{S_t, \hat{S}_t\}$.

2.1. Equivalence and simplification of team forms

In general, Problem 1 is computationally intractable. For discrete valued random variables, the problem belongs in NEXP complexity class (Goldman & Zilberstein, 2004). For continuous valued random variables, even simple two stage systems like the Witsenhausen counterexample (Witsenhausen, 1968) has remained an open problem for decades. As such, most research in team theory has focused on identifying specific information structures that are amenable to analysis. See Mahajan et al. (2012) for an overview.

In this paper, we focus on an intermediate step involved in solving Problem 1. For that matter, we define the following.

Definition 4 (*Sufficient and Irrelevant Information for Control*)**.** The data $J_\alpha \subseteq I_\alpha, \alpha \in A$, is said to be *sufficient information for control* at agent α if restricting attention to control laws of the form

$$g_\alpha : \left(X_{J_\alpha}, \mathscr{F}_{J_\alpha}\right) \to (\mathbb{X}_\alpha, \mathscr{F}_\alpha)$$

in Problem 1 is without loss of optimality. The remaining data $I_\alpha \setminus J_\alpha$ is said to be *irrelevant information for control* at agent α.

For example, in the real-time communication system of Example 1, the data $(S_{1:t-1}, Q_{1:t-1})$ is irrelevant to the encoder at time t. Identifying such a simplified form of the controller is usually the first step towards a complete solution of team problems. For that reason, instead of Problem 1, we are interested in the following problem:

Problem 2 (*Sufficient Information for Control*)**.** Given a sequential team, identify sufficient information for control at all agents. Or equivalently, identify irrelevant information for control at all agents.

Definition 5 (*Equivalence of Two Team Forms*)**.** Two team forms $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\})\rangle$ and $\mathcal{F}' = \langle (N', A', B', M'), \{I'_n : n \in A' \cup M'\}, (K' : \{D'_k, k \in K'\})\rangle$ are *equivalent* if

(1) They have the same variable and cost structures, i.e., there exist a permutation σ of N and a permutation π of K such that $(\sigma(N), \sigma(A), \sigma(B), \sigma(M)) = (N', A', B', M'), \pi(K) = K'$, and $\forall k \in K, \sigma(D_k) = D'_{\pi(k)}$.
(2) The information sets of the system dynamics are identical, i.e., $\forall m \in M, I_m = I'_{\sigma(m)}$.
(3) For *any choice* of team types $\mathcal{T}$, the values of the teams $(\mathcal{F}, \mathcal{T})$ and $(\mathcal{F}', \mathcal{T})$ are the same.

In the remainder of this paper, we will assume that the permutations σ and π are identity transformations.

Definition 6 (*Simplification of a Team Form*)**.** A team form $\mathcal{F}' = \langle (N', A', B', M'), \{I'_n : n \in A' \cup M'\}, (K' : \{D'_k, k \in K'\})\rangle$ is a *simplification* of a team form $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\})\rangle$ if

(1) $\mathcal{F}$ and $\mathcal{F}'$ are equivalent; and
(2) for all $\alpha \in A, I'_\alpha \subseteq I_\alpha$ and at least one of the inclusions is strict.

Definition 7 (*Minimal Simplification*)**.** A team form $\mathcal{F}'$ is a *minimal simplification* of $\mathcal{F}$ if

(1) $\mathcal{F}'$ is a simplification of $\mathcal{F}$; and
(2) there does not exist a team form $\mathcal{F}''$ such that $\mathcal{F}''$ is a simplification of $\mathcal{F}'$.

Note that the simplification of team forms defines a partial order on the set of all equivalent team forms; hence, a minimal team form need not be unique.

Identifying irrelevant information (Problem 2) at all agents is equivalent to identifying a simplification of the corresponding team form. In particular, in the above definitions, the data $I_\alpha \setminus I'_\alpha$ is irrelevant at agent α. Hence, Problem 2 is equivalent to the following problem:

Problem 3. Given a team form, identify one of its minimal simplifications.

3. Preliminaries on graphical models

In the sequel we model the team form as a DAG (directed acyclic graph). Simplification of a team form may be viewed as removing edges from this DAG using graphical modeling algorithms. We present background on graphical models in this section.

3.1. Directed graphs

A directed graph G is a pair (V, E) where V is the set of vertices and $E \subset V \times V$ is the set of edges. An edge (u, v) in E is considered directed from u to v; u is the *in-neighbor* or *parent* of v; v is the *out-neighbor* or *child* of u; and u and v are *neighbors*. The set of in-neighbors of v, called the *in-neighborhood* of v is denoted by $N_G^-(v)$; the set of out-neighbors of u, called the *out-neighborhood* of u is denoted by $N_G^+(u)$; the set of neighbors of u, called the *neighborhood* of u, is denoted by $N_G(u)$.

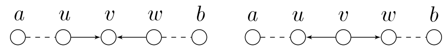
A *path* is a sequence of vertices such that each vertex has a directed edge to the next vertex in the sequence. The first vertex of a path is its *start node*, the last node is its *end node*. A *cycle* is a path with the same start and end node.

A *directed acyclic graph* (DAG) is a directed graph with no cycles. In a DAG, the set of all vertices u such that there is a path from u to v is called the *lower set* or *ancestors* of v and denoted by $\overleftarrow{v}$. Similarly, the set of all vertices v such that there is a path from u to v is called the *upper set* or *descendants* of u and denoted by $\overrightarrow{u}$. For a subset U of vertices, the lower set (or ancestral set) of U is given by $\overleftarrow{U} = \bigcup_{u \in U} \overleftarrow{u}$ and the upper (or descendant) set of u is given by $\overrightarrow{U} = \bigcup_{u \in U} \overrightarrow{u}$.

3.2. Bayesian networks with deterministic nodes

The terminology and definitions here are taken from Geiger, Verma, and Pearl (1990). A Bayesian network with deterministic nodes (V, E, D) is a DAG $G(V, E)$ and a subset D of vertices that are a deterministic function of their parents. A joint distribution $\mathbb{P}$ over $(X_v : v \in V)$ is said to *deterministically factor* with respect to (V, E, D) if

$$\mathbb{P}(X_V) = \bigotimes_{v \in V} \mathbb{P}(X_v | X_{N_G^-(v)}).$$



(a) A head-to-head vertex v in a trail from a to b.

(b) A tail-to-tail vertex v in a trail from a to b.

Fig. 1. Head-to-head and tail-to-tail vertices in a trail.

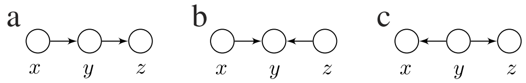


Fig. 2. An example illustrating the relation between D-separation and conditional independence.

Definition 8 (*Graphical Irrelevant*)**.** Given a Bayesian network (V, E, D) and sets $S_1, S_2, S_3 \in V$, S_1 is *irrelevant* to S_3 given S_2, denoted by $(S_1 \perp_{(V,E,D)} S_3)|S_2$, if for any joint measure $\mathbb{P}$ that recursively factors with respect to (V, E, D)

$$\mathbb{P}(X_{S_1}|X_{S_2}, X_{S_3}) = \mathbb{P}(X_{S_1}|X_{S_2}), \quad \mathbb{P}\text{-a.s.}$$

This conditional irrelevance can be expressed in terms of a graph property called D-separation. To define D-separation, we need some notation and terminology from graphical models.

A *trail* is a sequence of vertices such that each vertex is a neighbor of the next vertex in the sequence. A vertex v is called a *head-to-head* vertex with respect to a trail t, if t contains consecutive vertices uvw such that $(u, v), (w, v) \in E$. See Fig. 1(a). A vertex v is called a *tail-to-tail* vertex with respect to a trail t, if t contains consecutive vertices uvw such that $(v, u), (v, w) \in E$. See Fig. 1(b). A vertex that starts or ends a trail is a tail-to-tail vertex if it has an outgoing edge but it is *not* a head-to-head vertex if it has an incoming edge.

Definition 9 (*Functionally Determined Vertices*)**.** A vertex v is functionally determined by $S \subset V$ iff v is in S or v is a deterministic vertex and all its parents are functionally determined by S. A set of vertices is functionally determined by S if each of its members is functionally determined by S.

Definition 10 (*D-Separation*)**.** Given a Bayesian network (V, E, D) and disjoint sets $S_1, S_2, S_3 \in V$, S_2 is said to *D-connect* S_1 to S_3 iff there exists a trail t between a vertex in S_1 and a vertex in S_3 such that

(1) Every head-to-head vertex in t is in S_2 or has a descendant in S_2.
(2) No non head-to-head vertex in t is functionally determined by S_2.

If S_2 does not D-connect S_1 to S_3, then S_2 is said to *D-separate* S_1 from S_3.

A standard result in Bayesian networks is the following (Geiger et al., 1990; Shachter, 1998).

Proposition 1. *Given a Bayesian network (V, E, D) and disjoint sets $S_1, S_2, S_3 \in V$, we have $(S_1 \perp_{(V,E,D)} S_3)|S_2$ iff S_2 D-separates S_1 from S_3.*

As an example, consider the three graphical models shown in Fig. 2.

- In case (a), $P(x, y, z) = P(x)P(y|x)P(z|y)$; hence x and z are conditionally independent given y. Note that y D-separates x from z.
- In case (b), $P(x, y, z) = P(x)P(z)P(y|x, z)$; hence x and z are independent but they are not *conditionally* independent given y. Note that y does *not* D-separate x from z.

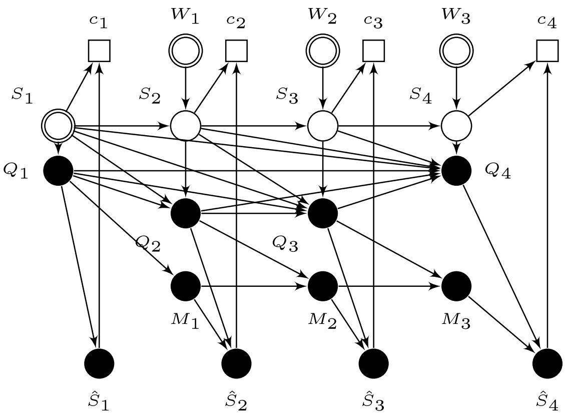


Fig. 3. A DAG corresponding to the team form of the real-time communication system of Example 1.

- In case (c), $P(x, y, z) = P(y)P(x|y)P(z|y)$; hence x and z are conditionally independent given y. Note that y D-separates x from z.

D-separation can be checked in $O(|E|)$ time by either using an edge labeling variation of breadth first search (Geiger et al., 1990) or using an edge based reachability analysis (called Bayes Ball) (Shachter, 1998).

4. Representing a team form as a DAG

A team form may be modeled as a DAG as follows.

- *Vertices*
  (1) Represent each control variable X_α, $\alpha \in A$, by a vertex marked with a full circle ● and labeled with X_α. Call the collection of all such vertices as V_A.
  (2) Represent each primitive variable X_β, $\beta \in B$, by a vertex marked with a double circle ⦾ and labeled with X_β. Call the collection of all such vertices as V_B.
  (3) Represent each system variable X_m, $m \in M$, by a vertex marked with an empty circle ○ and labeled with X_m. Call the collection of all such vertices as V_M.
  (4) Represent each cost term c_k, $k \in K$, by a vertex marked with a square □ and labeled with c_k. Call the collection of all such vertices as V_K.
  Thus, the vertex set of the DAG is
  $$V = V_A \cup V_B \cup V_M \cup V_K.$$
- *Edges*
  (1) For all $n \in A \cup M$ and $m \in I_n$ draw an edge from X_m to X_n.
  (2) For all $k \in K$ and $n \in D_k$ draw an edge from X_n to c_k.
  Thus, the edge set E of the DAG is
  $$E = \Big(\bigcup_{n\in A\cup M}\bigcup_{m\in I_n}(X_m, X_n)\Big) \cup \Big(\bigcup_{k\in K}\bigcup_{n\in D_k}(X_n, c_k)\Big).$$

For example, the DAG corresponding the real-time communication system of Example 1 is shown in Fig. 3.

4.1. Consistency of graphical representation

For any choice of a team type $\mathcal{T}$ and a control policy $\mathbf{g}$, the joint probability measure $P_{\mathbf{g}}$ on $(X_n : n \in N)$ factors according to (1). This factorization is of the form

$$\mathbb{P}(X_N) = \bigotimes_{n\in N} \mathbb{P}(X_n|X_{N_G^-(n)})$$

where for $n \in A \cup M$, X_n is a deterministic function of $\{X_m : m \in N_G^-(n)\}$. Thus, the joint probability *deterministically factors*

Algorithm 1: StrictlyExpand

```
input : Team form $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\}) \rangle$
output: Team form $\mathcal{F}'$
begin
    forall $\alpha_\circ \in A$ do
        forall $n_\circ \in A \cup M$ do
            if $I_{n_\circ} \subset I_{\alpha_\circ}$ and $X_{n_\circ} \notin I_{\alpha_\circ}$ then
                let $I'_{\alpha_\circ} := I_{\alpha_\circ} \cup \{X_{n_\circ}\}$
                $I_{\alpha_\circ} \leftarrow I'_{\alpha_\circ}$
                return StrictlyExpand($\mathcal{F}$)
    return $\mathcal{F}$
```

according to the DAG $G(V, E)$ with $(A \cup M)$ deterministic nodes. Hence, the DAG representation of a sequential team is a Bayesian network (Geiger et al., 1990).

4.2. Simplification of team forms

Simplification of a team form has a nice graphical interpretation. Let G and G' be the DAG corresponding to $\mathcal{F}$ and $\mathcal{F}'$. If $\mathcal{F}'$ is a simplification of $\mathcal{F}$, then we can obtain G' from G by dropping some of the incoming edges to vertices in V_A (marked by full circle ●). To find the minimal simplification of a team form, we describe an iterative procedure, where we drop some of the incoming edges to vertices in V_A at each step. When this procedure cannot find any edges to remove, the resultant team form is minimal.

5. Simplification at a single agent

In this section we present algorithms to remove irrelevant data from a single agent, which is used in the next section to develop algorithms to remove irrelevant data from a group of agents.

Before proceeding, we need to take care of a technicality. Given a team form $\mathcal{F}$, it is sometimes possible to find an equivalent team form $\mathcal{F}'$ such

$$\forall \alpha \in A, \quad |I'_\alpha| \le |I_\alpha| \text{ but } I'_\alpha \not\subseteq I_\alpha.$$

To avoid such situations, we always start with a strict expansion of the information structure of a team form.

5.1. Strict expansion of an information structure

The idea of a strict expansion of an information structure was introduced in Witsenhausen (1975). An information structure is strictly expanded if whenever an agent α knows the information set of a control or system variable then it also knows the corresponding control or system variable. Formally, we have the following:

Definition 11 (*Strictly Expanded Information Structure*)**.** An information structure $\{I_n : n \in A \cup M\}$ is *strictly expanded* if for any $n \in A \cup M$ and $\alpha \in A$ such that $I_n \subset I_\alpha$, we have that $X_n \in I_\alpha$.

A team form can be strictly expanded by a simple iterative procedure shown in Algorithm 1. The algorithm always converges in a finite number of steps because N is finite. Thus, in the sequel we make the following assumption:

Assumption (A1). The information structure of the team form is strictly expanded.

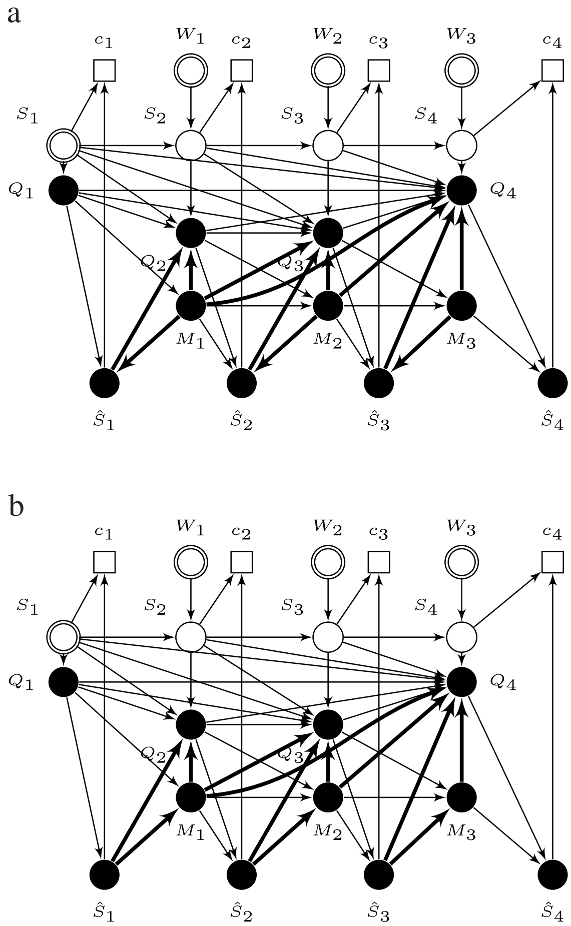


Fig. 4. Two possible strict expansions of information structure of the team form of Example 1. The thick lines denote the edges added as part of expanding the information structure.

The information structure of Example 1 is not strictly expanded. Its strict expansion using Algorithm 1 is not unique and depends on the order in which we proceed. Two possible strict expansions of the team form of Fig. 3 are shown in Fig. 4.

The multiplicity of strict expansions is not a concern because (i) the operation of strictly expanding a team form is idempotent; and (ii) all strict expansions of a team form are equivalent. The first property follows from construction and the second follows from the proposition given below.

Proposition 2. *Given any team form $\mathcal{F}$, any strict expansion of the information structure of $\mathcal{F}$ results in a team form that is equivalent to $\mathcal{F}$.*

See Appendix A for proof.

5.2. Main result for single agent

Definition 12 (*Dependent Cost*)**.** For any agent $\alpha \in A$, let $K_\alpha \subset K$ denote the cost terms that are influenced by the control actions X_α, that is,

$$K_\alpha = \{k \in K : D_k \cap \overrightarrow{X}_\alpha \neq \emptyset\}.$$

Theorem 2 (*Irrelevant Information at Single Agent*)**.** *For $\alpha \in A$ and $J_\alpha \subset I_\alpha$, the data $\{X_n; n \in J_\alpha\}$ is irrelevant for control at agent α if, in the graphical model corresponding to the team form, $I_\alpha \setminus J_\alpha \cup \{X_\alpha\}$ D-separates J_α from K_α.*

```
Algorithm 2: SimplifyAtNode
  input  : Team form $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\}) \rangle$
  input  : Agent $\alpha$
  output : Team form $\mathcal{F}'$
  output : Irrelevant data $J_\alpha$
  begin
      let $K_\alpha := \{k \in K : D_k \cap \overrightarrow{X}_\alpha \neq \emptyset\}$
      let $J_\alpha :=$ GraphIrrelevant($\mathcal{F}, I_\alpha \cup \{\alpha\}, K_\alpha$)
      $I_\alpha \leftarrow I_\alpha \setminus J_\alpha$
      return $(\mathcal{F}, J_\alpha)$
```

Proof. Arbitrarily fix the control policy $\mathbf{g}_{-\alpha} = \{g_{\alpha'} : \alpha' \in A \setminus \{\alpha\}\}$ of agents except agent α.

For any $K' \subset K$, define $C(K') = \sum_{k \in K'} c(X_{D_k})$ and decompose the total cost as $C(K) = C(K_\alpha) + C(K \setminus K_\alpha)$. Note that $C(K \setminus K_\alpha)$ does not depend on X_α and hence on g_α. Thus,

$$\arg\min_{g_\alpha} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K)] = \arg\min_{g_\alpha} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha)] \tag{2}$$

which implies that *to choose an optimal* g_α, agent α only needs to consider the cost terms K_α.

Let $S_\alpha = I_\alpha \setminus J_\alpha$. Suppose J_α is such that (S_α, α) D-separates J_α from K_α. An immediate consequence of this D-separation is that

$$\mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha) | X_{I_\alpha}, X_\alpha] = \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha) | X_{S_\alpha}, X_\alpha]. \tag{3}$$

Now, we can combine (2) and (3) with Theorem 1 to complete the proof as follows: Let

$$h(X_\alpha, X_{S_\alpha}) = \mathbb{E}_{\mathbf{g}_{-\alpha}}[C(K_\alpha) | X_{S_\alpha}, X_\alpha].$$

Eq. (2) implies that agent α is interested in minimizing

$$\begin{aligned}
\mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha)] &= \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}\big[\mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha) | X_{I_\alpha}, X_\alpha]\big] \\
&\overset{(a)}{=} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}\big[\mathbb{E}_{\mathbf{g}_{-\alpha}}[C(K_\alpha) | X_{I_\alpha}, X_\alpha]\big] \\
&\overset{(b)}{=} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}\big[\mathbb{E}_{\mathbf{g}_{-\alpha}}[C(K_\alpha) | X_{S_\alpha}, X_\alpha]\big] \\
&\overset{(c)}{=} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[h(X_\alpha, X_{S_\alpha})] \\
&= \mathbb{E}_{\mathbf{g}_{-\alpha}}[h(g_\alpha(X_{S_\alpha}, X_{J_\alpha}), X_{S_\alpha})]
\end{aligned} \tag{4}$$

where (a) follows from policy independence of conditional expectation (Witsenhausen, 1975), (b) follows from (3), and (c) follows from the definition of $h(\cdot)$. Now, Theorem 1 implies that to minimize the RHS of (4), there is no loss of optimality in choosing X_α as

$$X_\alpha = g_\alpha(X_{S_\alpha}).$$

Hence, the data $\{X_n; n \in J_\alpha\}$ is irrelevant for control at agent α.

Theorem 2 suggests that given a team form $\mathcal{F}$ and an agent α, if we identify a subset $J_\alpha \subset I_\alpha$ such that $(I_\alpha \setminus J_\alpha) \cup \{\alpha\}$ D-separates J_α from the dependent cost K_α, then the team form obtained by removing the edges $\{(m, \alpha), m \in J_\alpha\}$ is a simplification of $\mathcal{F}$. This procedure is shown in Algorithm 2. Since this algorithm is an implementation of Theorem 2, the output team form is a simplification of the input team form.

In order to identify an appropriate subset J_α, any standard graphical modeling algorithm for checking D-separation (or identifying graphically irrelevant data) may be used. One such algorithm is the Bayes ball algorithm (Shachter, 1998). This algorithm, refined for the graphical model corresponding to the team form, is presented in Algorithm 7 in Appendix B. We refer to this algorithm as `GraphIrrelevant`.

```
Algorithm 3: SimplifyTeam (Initial Version)
  input  : Team form $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\}) \rangle$
  output : Team form $\mathcal{F}'$
  begin
      $\mathcal{F} \leftarrow$ StrictlyExpand($\mathcal{F}$)
      repeat
          let $\mathcal{F}' := \mathcal{F}$
          forall $\alpha \in A$ do
              $(\mathcal{F}, J) \leftarrow$ SimplifyAtNode($\mathcal{F}, \alpha$)
      until $\mathcal{F}' = \mathcal{F}$
```

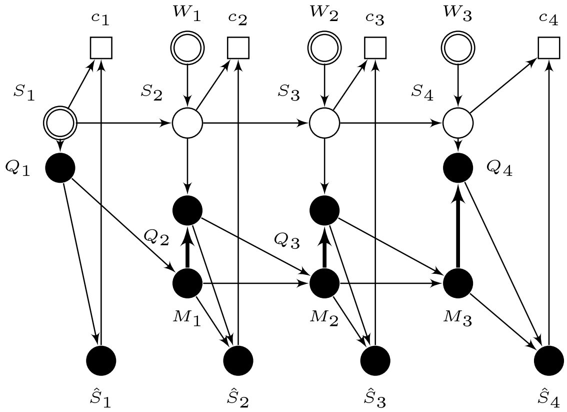


Fig. 5. Simplification of the team forms shown in Fig. 4. The thick lines denote the edges that were added in the strict expansion of the team form.

To simplify a team form, we can apply Algorithm 2 at all agents. We can iterate this step as long as we remove some edges at each iteration. This procedure is shown in Algorithm 3. Since each iteration simplifies the original team form, the final team form is also a simplification of the input team form.

Using Algorithm 3 on either of the team forms shown in Fig. 4 gives the team form shown in Fig. 5. The information set of Q_t is $\{S_t, M_{t-1}\}$, of M_t is $\{Q_t, M_{t-1}\}$, and of $\hat{S}_t$ is $\{Q_t, M_{t-1}\}$. Thus, the structure of optimal encoder, decoder, and memory update is

$$Q_t = e_t(S_t, M_{t-1}), \qquad \hat{S}_t = d_t(Q_t, M_{t-1}), \qquad M_t = g_t(Q_t, M_{t-1})$$

which is equivalent to the result derived in Witsenhausen (1979).

6. Simplification at a group of agents

In this section we present algorithms to remove irrelevant data from a group of agents. Recursively using this step leads to a simplification of a team form. The key idea behind the algorithms presented in this section is to use the idea of a *coordinator* presented in Nayyar (2011) and Nayyar et al. (2013).

6.1. Common and local information

Given a subset H of agents in a team form $\mathcal{F}$, we can split the information sets of all agents $\alpha \in H$ into two parts: the *common information* $C_H := \bigcap_{\alpha \in H} I_\alpha$ and the *local information* (or private information) $L_{H,\alpha} := I_\alpha \setminus C_H$, $\alpha \in H$. The common information is the data known to all agents in H; in fact, it is *common knowledge* (in the sense of Aumann (1976)) to all agents in H. The local information at α is the data known at α but not known at *every* other agent in H.

Algorithm 4: CoordinatedTeam

```
input  : Team form $\mathscr{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\}) \rangle$
input  : Group of agents $H$
output: Coordinated team form $\mathscr{F}_H$
output: Coordinator $\lambda_H$
begin
    let $\lambda_H$ = new node
    let $(A^*, B^*, M^*, N^*) = (A \setminus H \cup \{\lambda_H\}, B, M \cup A \setminus H, N \cup \{\lambda_H\})$
    let $C_H = \bigcap_{\alpha \in H} I_\alpha$
    $I^*_{\lambda_H} \leftarrow C_H$
    forall $\alpha \in H$ do
        $I^*_\alpha \leftarrow (I_\alpha \cup \lambda_H) \setminus C_H$
    forall $n \in A \cup M \setminus H$ do
        $I^*_n \leftarrow I_n$
    $\mathscr{F}_H \leftarrow \langle (N^*, A^*, B^*, M^*), \{I^*_n : n \in A^* \cup M^*\}, (K, \{D_k : k \in K\}) \rangle$
    return $(\mathscr{F}_H, \lambda_H)$
```

6.2. Coordinator for a group of agents

Given a team form $\mathscr{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\}) \rangle$ and a subset H of agents, construct a *coordinated team form* $\mathscr{F}_H = \langle (N^*, A^*, B^*, M^*), \{I^*_n : n \in A^* \cup M^*\}, (K^*, \{D^*_k : k \in K^*\}) \rangle$ with the following components:

(1) *Variable structure*: The coordinated team form $\mathscr{F}_H$ has a new agent, called the *coordinator* and denoted by λ_H; the agents in H are passive; and the system and primitive variables are unchanged. Thus,

$$A^* = (A \setminus H) \cup \{\lambda_H\}, \qquad M^* = M \cup H, \qquad B^* = B. \qquad (5)$$

(2) *Information structure*: The information set of the coordinator is the common information in H, i.e.,

$$I^*_{\lambda_H} = C_H;$$

the information set of $\alpha \in H$ is the local information $\{X_n : n \in L_{H,\alpha}\}$ and the coordinator's action $X^*_{\lambda_H}$, i.e.,

$$I^*_\alpha = L_{H,\alpha} \cup \{\lambda_H\}, \quad \alpha \in H;$$

and the information set of all other variables remains unchanged, i.e.,

$$I^*_n = I_n, \quad n \in A \cup M \setminus H.$$

(3) *Cost structure*: The cost structure of $\mathscr{F}_H$ is the same as that of $\mathscr{F}$.

The procedure for obtaining a coordinated team form is shown in Algorithm 4. In addition, for any team type $\mathscr{T}$ of $\mathscr{F}$, a team type $\mathscr{T}_H$ of $\mathscr{F}_H$ is obtained as follows:

(1) *Measurable spaces*: For $n \in N^* \setminus \{\lambda_H\}$, $(\mathbb{X}^*_n, \mathscr{F}^*_n) = (\mathbb{X}_n, \mathscr{F}_n)$. For $n = \lambda_H$, $X^*_{\lambda_H}$ is a function valued random variable. The measurable space $(\mathbb{X}^*_{\lambda_H}, \mathscr{F}^*_{\lambda_H})$ is given by

$$\mathbb{X}^*_{\lambda_H} = \prod_{\alpha \in H} \mathbb{Z}^\alpha_H$$

where $\mathbb{Z}^\alpha_H$, $\alpha \in H$, is the space of measurable functions from $(\mathbb{X}_{L_{H,\alpha}}, \mathscr{F}_{L_{H,\alpha}})$ to $(\mathbb{X}_\alpha, \mathscr{F}_\alpha)$ and $\mathscr{F}^*_{\lambda_H}$ is the corresponding Borel measure on the product of function spaces $\mathbb{Z}^\alpha_H$.

(2) *Probability measure* on primitive random variables remains unchanged, i.e., $Q^* = Q$.

(3) *System dynamics*: For $m \in M$, the system dynamics are not changed, i.e., $f^*_m = f_m$. For $m \in H$, the system dynamics are given by

$$f^*_m(X^*_{\lambda_H}, X^*_{L_{H,\alpha}}) = Z^\alpha_H(X^*_{L_{H,\alpha}})$$

where $X^*_{\lambda_H} = (Z^\alpha_H : \alpha \in H)$.

(4) *Cost functions*: The cost functions of $\mathscr{T}_H$ are the same as those of $\mathscr{T}$.

While modeling the coordinated team as a DAG, we will

(1) Represent the coordinator X_{λ_H}, by a vertex marked with full diamond ♦ and labeled with X_{λ_H} (or, when clear from context, by just λ_H).

(2) Represent all agents X_α, $\alpha \in H$ by a vertex marked with an empty circle ○ and labeled with X_α. Call the collection of all such vertices V_H.

(3) Represent the remaining variables, $N \setminus H$, as earlier.

By construction, the out-neighborhood of X_{λ_H} is V_H, i.e.,

$$N^+_G(X_{\lambda_H}) = V_H := \{X_\alpha : \alpha \in H\}.$$

The idea of a coordinator based on common information was proposed in Nayyar (2011) and Nayyar et al. (2013). The coordinated team form is useful for two reasons: (i) it is equivalent to the original team form (see Proposition 3), and hence (ii) any structural result for the coordinated team is also applicable to the original (see Theorem 3). Thus, using the coordinated team, we can extend any solution technique for identifying irrelevant data at a single agent to a solution technique for identifying irrelevant data at a group of agents.

Proposition 3. *Given any sequential team $(\mathscr{F}, \mathscr{T})$ and a subset H of agents, let $(\mathscr{F}_H, \mathscr{T}_H)$ be the corresponding coordinated sequential team. Then for any policy $\mathbf{g}$ of $(\mathscr{F}, \mathscr{T})$, there exists a policy $\mathbf{g}_H$ of $(\mathscr{F}_H, \mathscr{T}_H)$ that achieves the same cost and vice versa, i.e., for any policy $\mathbf{g}_H$ of $(\mathscr{F}_H, \mathscr{T}_H)$, there exists a policy $\mathbf{g}$ of $(\mathscr{F}, \mathscr{T})$ that achieves the same cost.*

This is proved in Appendix C. The manner in which the corresponding policies are constructed implies that if we prove that a subset J_H of C_H is irrelevant for the coordinator λ_H in $\mathscr{F}_H$, then J_H is also irrelevant for all $\alpha \in H$ in $\mathscr{F}$. Using this correspondence, we state the main result for a group of agents.

6.3. Main result for a group of agents

Theorem 3. *Given a team form $\mathscr{F}$ and a subset H of agents A, let $\mathscr{F}_H$ be the corresponding coordinated team. Then, for any $J_H \subset C_H$, the data $\{X_n : n \in J_H\}$ is irrelevant for control at all agents $\alpha \in H$ if, in the graphical model corresponding to $\mathscr{F}_H$, $C_H \setminus J_H \cup \{\lambda_H\}$ D-separates J_H from K_{λ_H}.*

Proof. Let $S_H = C_H \setminus J_H$. If (S_H, X_{λ_H}) D-separates J_H from K_{λ_H} in the graphical model corresponding to $\mathscr{F}_H$, then, by Theorem 2, the data $\{X_n : n \in J_H\}$ is irrelevant for control at the coordinator in $\mathscr{F}_H$. By the manner in the which the equivalent policy was constructed in the proof of Proposition 3, the data $\{X_n : n \in J_H\}$ is irrelevant at all agents in $\alpha \in H$ in $\mathscr{F}$.

Theorem 3 suggests that to simplify a team form $\mathscr{F}$ at a group of agents H, we should look at the coordinated team $\mathscr{F}_H = \texttt{CoordinatedTeam}(\mathscr{F}, H)$ and identify the irrelevant data J_H at the coordinator λ_H (using, for example, Algorithm 7). Then the team form obtained by removing the edges $\{(m, \alpha), m \in J_H, \alpha \in H\}$ is a simplification of $\mathscr{F}$. This procedure is implemented in Algorithm 5.

To simplify a team form, we can apply Algorithm 5 at all subsets of agents. We can iterate this step as long as we remove some edges

```
Algorithm 5: SimplifyAtGroup
    input  : Team form $\mathscr{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\})\rangle$
    input  : Group of agents $H$
    output : Team form $\mathscr{F}'$
    begin
        let $(\mathscr{F}_H, \lambda_H) :=$ CoordinatedTeam($\mathscr{F}, H$)
        $(\mathscr{F}_H, J_H) \leftarrow$ SimplifyAtNode($\mathscr{F}_H, \lambda_H$)
        forall $\alpha \in H$ do
            $I_\alpha \leftarrow I_\alpha \setminus J_H$
        return $\mathscr{F}$
```

```
Algorithm 6: SimplifyTeam (Final Version)
    input  : Team form $\mathscr{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\})\rangle$
    output : Team form $\mathscr{F}'$
    begin
        $\mathscr{F} \leftarrow$ StrictlyExpand($\mathscr{F}$)
        repeat
            let $\mathscr{F}' := \mathscr{F}$
            forall $H$ subset of $A$ do
                $\mathscr{F} \leftarrow$ SimplifyAtGroup($\mathscr{F}, H$)
        until $\mathscr{F}' = \mathscr{F}$
```

at each iteration. This procedure is shown in Algorithm 6. Since each iteration simplifies the original team form, the final team form is also a simplification of the input team form.

Remark 2. (1) In Algorithm 6, we do not need to check all subsets of agents. Rather, we only need to check subsets H for which the common information $C_H \neq \emptyset$. Hence, instead of iterating for all subsets of A in Algorithm 6, we only need to iterate over subsets of A for which $C_H \neq \emptyset$.

(2) If for some $H \subset A$, $C_H = \emptyset$ then for any H' such that $H \subset H' \subset A$, $C_{H'} = \emptyset$. Thus, if we find any subset H for which $C_H = \emptyset$, we do not need to check H and any of its supersets. This can be incorporated in the function to generate the subsets of A.

7. Examples

We present two examples that illustrate the algorithmic procedure to remove irrelevant information.

7.1. Hierarchical control system

Consider a hierarchical control system that consists of three interconnected subsystems. Controller 1 influences subsystems 1, 2, and 3; controller 2 influences subsystems 2 and 3; and controller 3 influences subsystem 3.

Formally, let S^i_t denote the state of subsystem i and U^i_t denote the control action of controller i. The initial states (S^1_1, S^2_1, S^3_1) are independent of each other and evolve according to

$$S^1_{t+1} = f^1_t(S^1_t, U^1_t, W^1_t), \qquad S^2_{t+1} = f^2_t(S^2_t, U^1_t, U^2_t, W^2_t),$$
$$S^3_{t+1} = f^3_t(S^3_t, U^1_t, U^2_t, U^3_t, W^2_t).$$

The information structures of the controllers are:

$$I^1_t = \{S^1_{1:t}, U^1_{1:t-1}\}, \qquad I^2_t = \{S^2_{1:t}, U^1_{1:t-1}, U^2_{1:t-1}\},$$
$$I^3_t = \{S^3_{1:t}, U^1_{1:t-1}, U^2_{1:t-1}, U^3_{1:t-1}\}.$$

The control actions are chosen as follows:

$$U^i_t = g^i_t(I^i_t), \quad i \in \{1, 2, 3\}.$$

The per-step cost is $c(\mathbf{S}_t, \mathbf{U}_t)$ where $\mathbf{S}_t = (S^1_t, S^2_t, S^3_t)$ and $\mathbf{U}_t = (U^1_t, U^2_t, U^3_t)$.

The optimization objective is to choose the control strategies $\mathbf{g}^i = (g^i_1, \dots, g^i_T)$, $i \in \{1, 2, 3\}$ to minimize the expected cost $\mathbb{E}_{(\mathbf{g}^1, \mathbf{g}^2, \mathbf{g}^3)}\left[\sum_{t=1}^T c(\mathbf{S}_t, \mathbf{U}_t)\right]$.

The team form corresponding to the above system is given by:

(1) *Variable structure*
- Control variables: $X_A = (U^1_{1:T}, U^2_{1:T}, U^3_{1:T})$.
- Primitive variables: $X_B = (S^1_1, S^2_1, S^3_1, W^1_{1:T}, W^2_{1:T}, W^3_{1:T})$.
- System variables: $X_M = (S^1_{2:T}, S^2_{2:T}, S^3_{2:T})$.
- All variables $X_N = X_A \cup X_B \cup X_M$.

(2) *Information structure*
- For any $X_n = U^1_t \in X_M, I_n = \{S^1_{1:t}, U^1_{1:t-1}\}$.
- For any $X_n = U^2_t \in X_M, I_n = \{S^2_{1:t}, U^1_{1:t-1}, U^2_{1:t-1}\}$.
- For any $X_n = U^3_t \in X_M, I_n = \{S^3_{1:t}, U^1_{1:t-1}, U^2_{1:t-1}, U^3_{1:t-1}\}$.
- For any $X_n = S^1_t \in X_M, I_n = \{S^1_{t-1}, U^1_{t-1}\}$.
- For any $X_n = S^2_t \in X_M, I_n = \{S^2_{t-1}, U^1_{t-1}, U^2_{t-1}\}$.
- For any $X_n = S^3_t \in X_M, I_n = \{S^3_{t-1}, U^1_{t-1}, U^2_{t-1}, U^3_{t-1}\}$.

(3) *Cost structure*
- Set of coupled costs $K = \{1, \dots, T\}$.
- For $t \in K, D_t = \{\mathbf{S}_t, \mathbf{U}_t\}$.

The DAG corresponding to the above team form for horizon $T = 4$ is shown in Fig. 6. For clarity, we have split the graph into two parts. The first graph shows the information sets of control variables and the second graph shows cost structure and the information sets of system variables.

Let $\mathscr{F}_0$ denote the team form of Fig. 6. Using Algorithm 6 on the above DAG proceeds as follows:

(1) $\mathscr{F}_1 =$ SimplifyAtNode($\mathscr{F}_0, U^3_4$). This removes the edges between $\{S^3_1, S^3_2, S^3_3\}$ and U^3_4. The resultant team form is shown in Fig. 7(a).

(2) $\mathscr{F}_2 =$ SimplifyAtNode($\mathscr{F}_1, U^2_4$). This removes the edges between $\{S^2_1, S^2_2, S^2_3\}$ and U^2_4.

(3) $\mathscr{F}_3 =$ SimplifyAtNode($\mathscr{F}_2, U^1_4$). This removes the edges between $\{S^1_1, S^1_2, S^1_3\}$ and U^1_4.

(4) $\mathscr{F}_4 =$ SimplifyAtNode($\mathscr{F}_3, U^3_3$). This removes the edges between $\{S^3_1, S^3_2\}$ and U^3_3.

(5) $\mathscr{F}_5 =$ SimplifyAtNode($\mathscr{F}_4, U^2_3$). This removes the edges between $\{S^2_1, S^2_2\}$ and U^2_3. The resultant team form is shown in Fig. 7(b).

(6) $\mathscr{F}_6 =$ SimplifyAtNode($\mathscr{F}_5, U^1_3$). This removes the edges between $\{S^1_1, S^1_2\}$ and U^1_3.

(7) $\mathscr{F}_7 =$ SimplifyAtNode($\mathscr{F}_6, U^3_2$). This removes the edges between $\{S^3_1\}$ and U^3_2. The resultant team form is shown in Fig. 7(c).

(8) $\mathscr{F}_8 =$ SimplifyAtNode($\mathscr{F}_7, U^2_2$). This removes the edges between $\{S^2_1\}$ and U^2_2.

(9) $\mathscr{F}_9 =$ SimplifyAtNode($\mathscr{F}_8, U^1_2$). This removes the edges between $\{S^1_1\}$ and U^1_2. The resultant team form is shown in Fig. 7(d).

In the team form $\mathscr{F}_9$, shown in Fig. 7(d), the information set of U^1_t is $\{S^1_t, U^1_{1:t-1}\}$; of U^2_t is $\{S^2_t, U^1_{1:t-1}, U^2_{1:t-1}\}$; and of U^3_t is $\{S^3_t, U^1_{1:t-1}, U^2_{1:t-1}, U^3_{1:t-1}\}$. Therefore, the structure of an optimal controller is

$$U^1_t = g^1_t(S^1_t, U^1_{1:t-1}), \qquad U^2_t = g^2_t(S^2_t, U^1_{1:t-1}, U^2_{1:t-1}),$$
$$U^3_t = g^3_t(S^3_t, U^1_{1:t-1}, U^2_{1:t-1}.U^3_{1:t-1}).$$

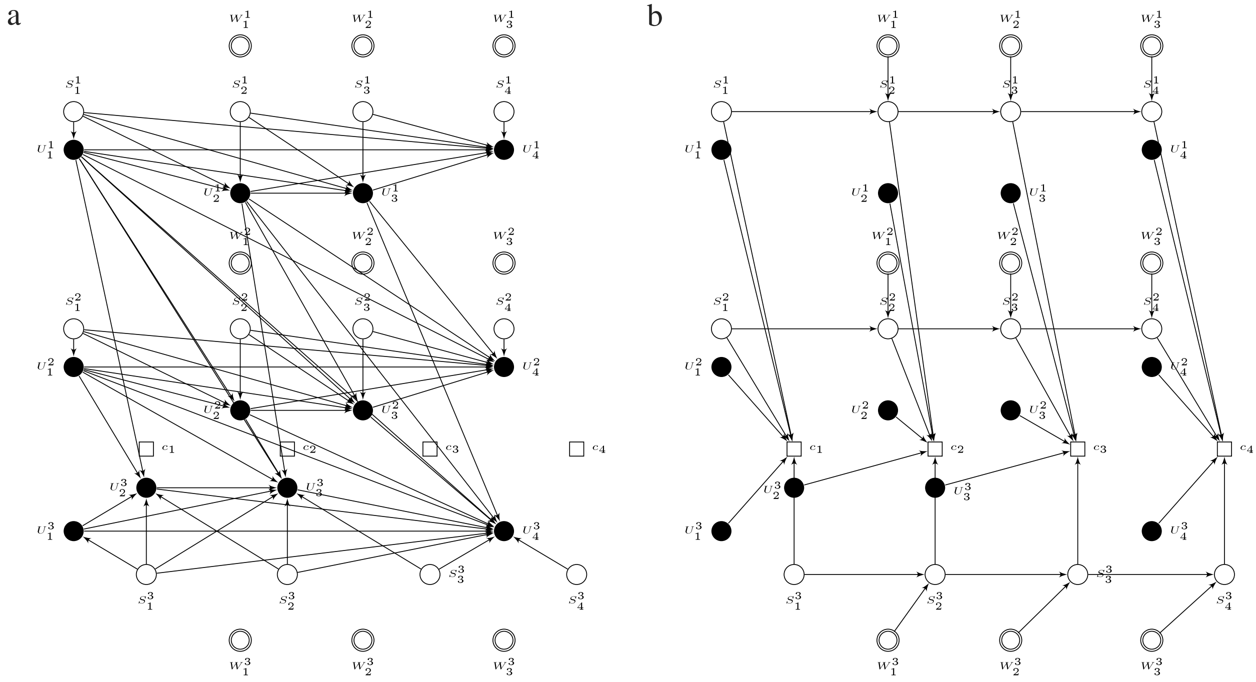


Fig. 6. A directed acyclic graph corresponding to the team form of a hierarchical control system. For clarity, we have split the edges into two graphs. The first graph shows the information sets of the control variables while the second graph shows the cost structure and the information sets of the system variables.

7.2. One-step delayed state sharing

Consider two coupled subsystems with one step delayed state sharing information structure (Aicardi, Davoli, & Minciardi, 1987). The state S_t^i of subsystem i, $i = 1, 2$, evolves according to

$$S_{t+1}^i = f_t^i(\mathbf{S}_t, \mathbf{U}_t, W_t^i)$$

where $\mathbf{S}_t = (S_t^1, S_t^2)$, $\mathbf{U}_t = (U_t^1, U_t^2)$, U_t^i denotes the control action of controller i and W_t^i is the process noise. The processes $\{W_t^1\}_{t=1}^T$ and $\{W_t^2\}_{t=1}^T$ are i.i.d. and independent of each other and of (S_1^1, S_1^2).

Each controller observes the state of its subsystem and the one-step delayed state of the other subsystem; it chooses a control action according to

$$U_t^i = g_t^i(S_t^i, \mathbf{S}_{1:t-1}).$$

The optimization problem is to choose the control strategies $\mathbf{g}^1 = (g_1^1, \ldots, g_T^1)$ and $\mathbf{g}^2 = (g_1^2, \ldots, g_T^2)$ to minimize the expected cost

$$\mathbb{E}_{(\mathbf{g}^1,\mathbf{g}^2)}\Big[\sum_{t=1}^{T} c(\mathbf{S}_t, \mathbf{U}_t)\Big].$$

The team form corresponding to the above system is given by:

(1) *Variable structure*
- Control variables: $X_A = (U_{1:T}^1, U_{1:T}^2)$.
- Primitive variables: $X_B = (S_1^1, S_1^2, W_{1:T}^1, W_{1:T}^2)$.
- System variables: $X_M = (S_{2:T}^1, S_{2:T}^2)$.
- All variables $X_N = X_A \cup X_B \cup X_M$.

(2) *Information structure*
- For any $X_n = U_t^i \in X_A$, $I_n = \{S_t^i, \mathbf{S}_{1:t-1}\}$.
- For any $X_n = S_t^i \in X_M$, $I_n = \{\mathbf{S}_{t-1}, \mathbf{U}_{t-1}\}$.

(3) *Cost structure*
- Set of coupled costs $K = \{1, \ldots, T\}$.
- For $t \in K$, $D_t = \{\mathbf{S}_t, \mathbf{U}_t\}$.

The DAG corresponding to the above team form for horizon $T = 4$ is shown in Fig. 8. For clarity, we have split the graph into two parts. The first graph shows the information sets of control variables and the second graph shows cost structure and the information sets of system variables.

Let $\mathcal{F}_0$ denote the team form of Fig. 8. Using Algorithm 6 on the above DAG proceeds as follows:

(1) $\mathcal{F}_1 = \mathtt{StrictlyExpand}(\mathcal{F}_0)$. The strict expansion of $\mathcal{F}_0$ gives a team form $\mathcal{F}_1$ shown in Fig. 9(a).
(2) $\mathcal{F}_2 = \mathtt{SimplifyAtGroup}(\mathcal{F}_1, H_1)$ where $H_1 = \{U_4^1, U_4^2\}$. In the corresponding coordinated team form $\mathcal{F}_{H_1}$, which is shown in Fig. 9(b), the set $\{S_3^1, S_3^2, U_3^1, U_3^2, \lambda_{H_1}\}$ D-separates $J_{H_1} = \{S_{1:2}^1, S_{1:2}^2, U_{1:2}^1, U_{1:2}^2\}$ from $K_{H_1} = \{c_4\}$. Removing the edges from J_{H_1} to λ_{H_1} in $\mathcal{F}_{H_1}$ gives the team form shown in Fig. 9(c); removing them in $\mathcal{F}_1$ gives the team form $\mathcal{F}_2$ shown in Fig. 9(d).
(3) $\mathcal{F}_3 = \mathtt{SimplifyAtGroup}(\mathcal{F}_2, H_2)$ where $H_2 = \{U_3^1, U_3^2\}$. In the corresponding coordinated team form $\mathcal{F}_{H_3}$, which is shown in Fig. 9(e), the set $\{S_2^1, S_2^2, U_2^1, U_2^2, \lambda_{H_2}\}$ D-separates $J_{H_2} = \{S_1^1, S_1^2, U_1^1, U_1^2\}$ from $K_{H_3} = \{c_3, c_4\}$. Removing the edges from J_{H_2} to λ_{H_2} in $\mathcal{F}_{H_2}$ gives the team form shown in Fig. 9(f); removing them in $\mathcal{F}_2$ gives the team form $\mathcal{F}_3$ shown in Fig. 8(c).

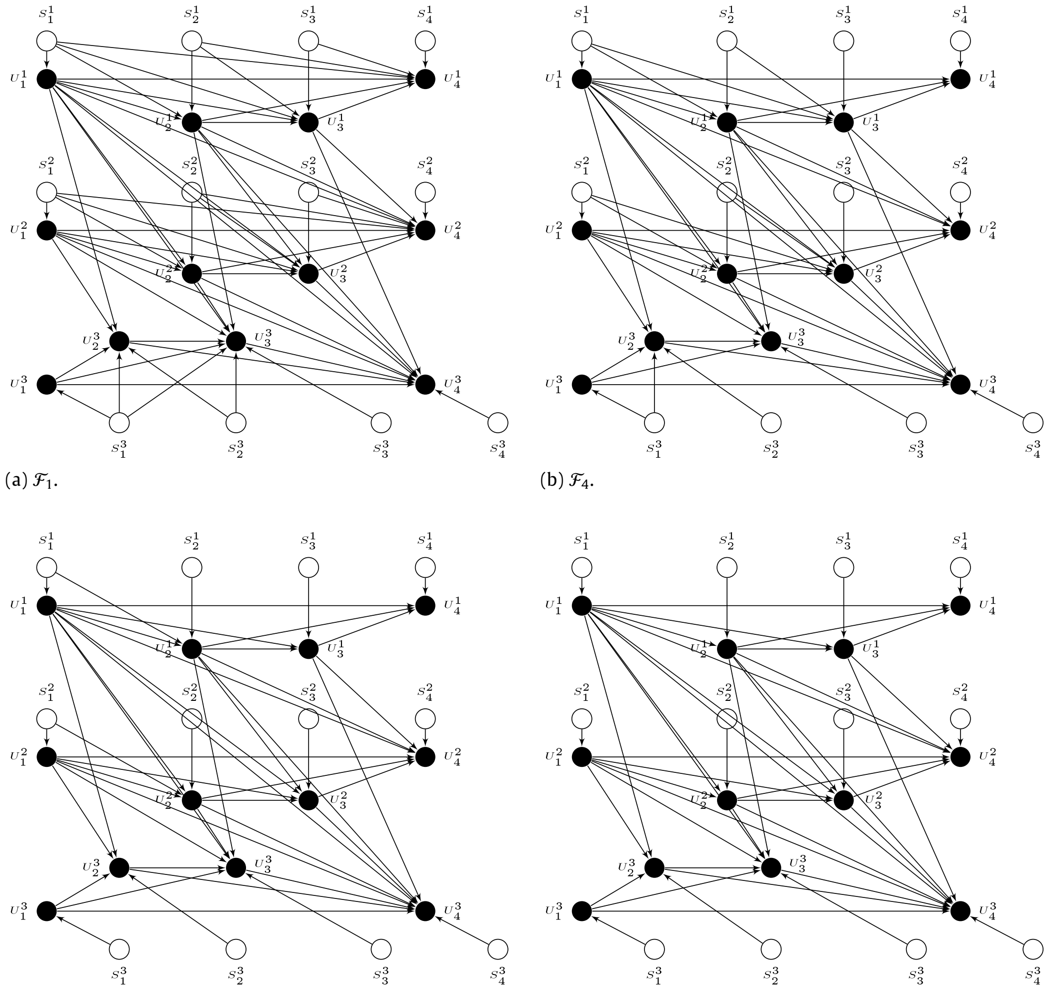
In the team form $\mathcal{F}_3$, shown in Fig. 8(c), the information set of U_t^i is $\{S_t^i, \mathbf{S}_{t-1}, \mathbf{U}_{t-1}\}$. Therefore, the structure of an optimal controller is

$$U_t^i = g_t^i(S_t^i, \mathbf{S}_{t-1}, \mathbf{U}_{t-1})$$

which is equivalent to the structural result derived in Aicardi et al. (1987).

8. Conclusion

In this paper, we present an algorithmic framework to identify irrelevant data in sequential teams. Our results depend only on the information structure and the cost coupling but do not depend on the specifics of the state spaces, probability measure, dynamics, and the cost function.



(a) $\mathcal{F}_1$.

(b) $\mathcal{F}_4$.

(c) $\mathcal{F}_7$.

(d) $\mathcal{F}_9$.

Fig. 7. Intermediate team forms obtained in the simplification of the team form of Fig. 6.

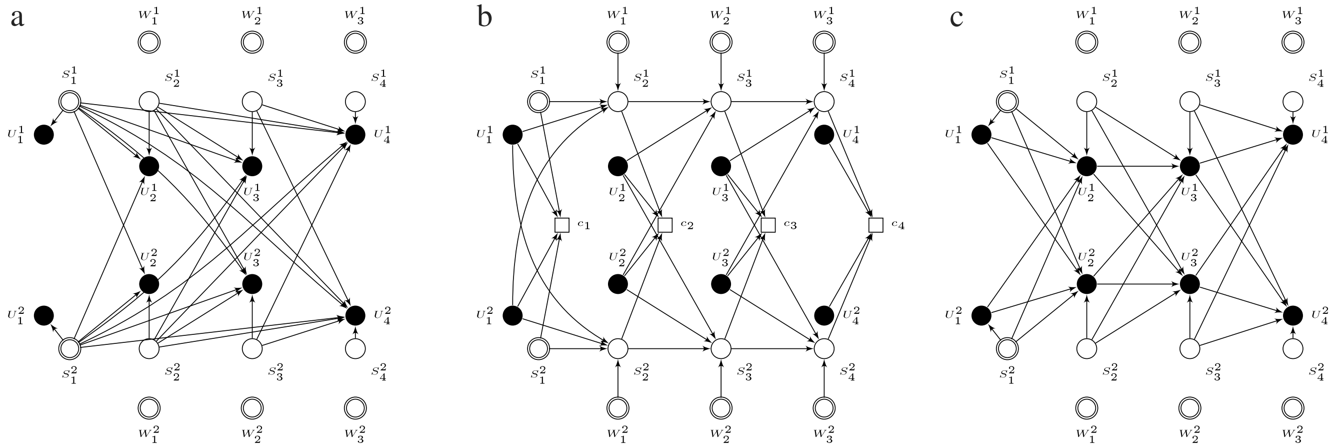
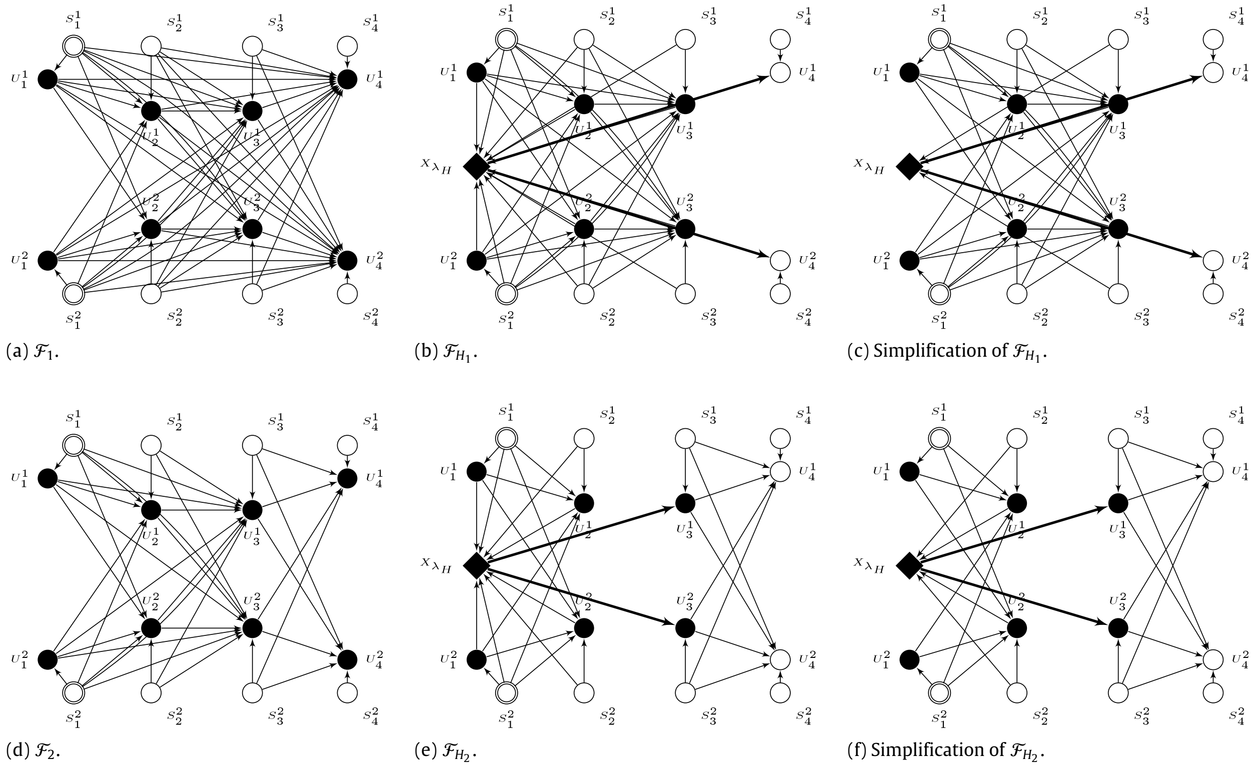


Fig. 8. A directed acyclic graph corresponding to the team form of decentralized control system with one-step delayed state sharing. For clarity, we have split the edges into two graphs: (a) the information sets of the control variables; (b) the cost structure and the information sets of the system variables. (c) shows the simplified team form obtained by using Algorithm 6.

The key idea behind the proposed algorithms is the following. A subset of data available to an agent is irrelevant only if appropriate conditional independence conditions hold. To algorithmically identify such irrelevant data, we model a sequential team using a graphical model. Then, checking for conditional independence is equivalent to checking D-separation in the corresponding graphical model, which may be verified using standard graphical modeling algorithms.

Once irrelevant data is identified, it may be removed without any loss of optimality. Removing data available to an agent is



(a) $\mathcal{F}_1$.

(b) $\mathcal{F}_{H_1}$.

(c) Simplification of $\mathcal{F}_{H_1}$.

(d) $\mathcal{F}_2$.

(e) $\mathcal{F}_{H_2}$.

(f) Simplification of $\mathcal{F}_{H_2}$.

Fig. 9. Intermediate team forms obtained in the simplification of the team form of Fig. 8.

equivalent to removing corresponding edges from the graphical model. We propose algorithms that sequentially remove edges from agents and groups of agents until no further edges can be removed.

The complexity of the proposed algorithm is proportional to the number of groups of agents that have non-empty common information. Thus, the algorithms run faster on systems where less data is shared between the agents i.e., the degree of decentralization is large. However, in the worst case, the complexity is exponential in the number of agents.

When we model a specific decentralized control system as a sequential team, as in Example 1 and Section 7, the number of agents increases linearly with time horizon. However, for the purpose of identifying irrelevant information, a short horizon (typically $T = 4$ or 5) is sufficient. This is because, effectively, we can use the steps of the algorithm to construct a backward induction based proof. For example, in the simplification presented in Fig. 9, the simplification at $T = 4$ may be thought of as the basis of induction, and the simplification at $T = 3$ and $T = 2$ may be thought of as induction steps. The horizon of the problem needs to large enough so that the “induction-step” is included in the simplification.

The algorithm described in this paper only identifies structural results that remove irrelevant information at the agents. Extending these algorithms to identify sufficient statistic of data available at the agents is an important future direction.

Acknowledgments

The authors are grateful to A. Anastasopoulos, A. Nayyar, M. Raginsky, D. Teneketzis, and S. Yüksel for discussions and comments. The authors thank the anonymous reviewers for valuable comments and suggestions to improve the presentation of the paper.

Appendix A. Proof of Proposition 2

Let $\mathcal{F}' = \texttt{StrictlyExpand}(\mathcal{F})$. Denote the series of team forms created by Algorithm 1 in going from $\mathcal{F}$ to $\mathcal{F}'$ by

$$\mathcal{F}_\ell = \{(N, A, B, M), \{I_n^\ell, n \in A \cup M\}, (K, \{D_k, k \in K\})\},$$

for $\ell = 0, \dots, L$, where $\mathcal{F}_0 = \mathcal{F}$, $\mathcal{F}_L = \mathcal{F}'$ and $\mathcal{F}_{\ell+1}$ is obtained by expanding $\mathcal{F}_\ell$ around (α_ℓ, n_ℓ), that is, for $\ell = 0, \dots, L$, $n \in A \cup M$,

$$I_n^{\ell+1} = \begin{cases} I^{\ell n}, & n \neq \alpha_\ell, \\ I_{\alpha_\ell} \cup \{X_{n_\ell}\}, & n = \alpha_\ell. \end{cases}$$

To show that $\mathcal{F}'$ is equivalent to $\mathcal{F}$ it suffices to prove that $\mathcal{F}_{\ell+1}$ is equivalent to $\mathcal{F}_\ell$, $\ell = 0, \dots, L-1$. Since $\mathcal{F}_\ell$ and $\mathcal{F}_{\ell+1}$ have the same variable and cost structure, they are equivalent if for any choice of team type $\mathcal{T}$, $(\mathcal{F}_\ell, \mathcal{T})$ and $(\mathcal{F}_{\ell+1}, \mathcal{T})$ have the same value.

By construction, $\mathcal{F}_\ell$ and $\mathcal{F}_{\ell+1}$ have the same variable and cost structure and the same information sets for system dynamics. Thus, to prove equivalence between $\mathcal{F}_\ell$ and $\mathcal{F}_{\ell+1}$ all we need to show is that for any choice of team type $\mathcal{T}$, the teams $(\mathcal{F}_\ell, \mathcal{T})$ and $(\mathcal{F}_{\ell+1}, \mathcal{T})$ have the value.

Consider a policy $\mathbf{g}' = \{g'_\alpha : \alpha \in A\}$ of $(\mathcal{F}_{\ell+1}, \mathcal{T})$. Since $I_{n_\ell} \subset I_{\alpha_\ell}, X_{n_\ell}$ may be written as $\psi(I_{\alpha_\ell}; \mathbf{g}')$ for some function ψ. Now, consider a policy $\mathbf{g} = \{g_\alpha : \alpha \in A\}$ of $(\mathcal{F}_\ell, \mathcal{T})$ such that

$$g_\alpha(I_\alpha) = \begin{cases} g'_\alpha(I_\alpha), & \alpha \neq \alpha_\ell \\ g'_{\alpha_\ell}(I_{\alpha_\ell}, \psi(I_{\alpha_\ell}; \mathbf{g}')), & \alpha = \alpha_\ell. \end{cases} \tag{A.1}$$

By construction, for any realization of the primitive random variables $\{X_\beta : \beta \in B\}$, using the policy $\mathbf{g}$ given by (A.1) in $(\mathcal{F}_\ell, \mathcal{T})$ leads to the same realization of system and control variables $(X_n : n \in A \cup M)$ as those by using policy $\mathbf{g}'$ in $(\mathcal{F}_{\ell+1}, \mathcal{T})$. Therefore, the value of $(\mathcal{F}_\ell, \mathcal{T})$ is at least as small as the value of $(\mathcal{F}_{\ell+1}, \mathcal{T})$. But by construction, any policy of $(\mathcal{F}_\ell, \mathcal{T})$ is also a policy of $(\mathcal{F}_{\ell+1}, \mathcal{T})$. Hence, the value of $(\mathcal{F}_{\ell+1}, \mathcal{T})$ is at least as small as the value of $(\mathcal{F}_\ell, \mathcal{T})$. Hence, $(\mathcal{F}_\ell, \mathcal{T})$ and $(\mathcal{F}_{\ell+1}, \mathcal{T})$ have the same value. Therefore, $\mathcal{F}_\ell$ and $\mathcal{F}_{\ell+1}$ are equivalent, and so are $\mathcal{F}_0 = \mathcal{F}$ and $\mathcal{F}_L = \mathcal{F}'$.

Algorithm 7: GraphIrrelevant (the Bayes Ball Algorithm)

```
input : Team form $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\})\rangle$
input : Observed data $O$
input : Target Set $T$
output: Irrelevant data $O'$
begin
    let $(V, S, R_*, R^*, P_*, P^*) = (\emptyset, T, T, \emptyset, N, N)$
    while $S \neq \emptyset$ do
        Pick $s \in S$
        $(V, S) \leftarrow (V \cup \{s\}, S \setminus \{s\})$
        if $s \in R_*$ then
            if $s \in O$ then
                // Do nothing. The ball is blocked
            else
                // Pass the ball through
                if $s \in P^*$ then
                    $P^* \leftarrow P^* \setminus \{s\}$
                    $S \leftarrow S \cup N^-_G(s)$
                if $s \in B \cap P_*$ then
                    $P_* \leftarrow P_* \setminus \{s\}$
                    $S \leftarrow S \cup N^+_G(s)$
        if $s \in R^*$ then
            if $s \in O$ then
                // Bounce back the ball
                if $s \in P^*$ then
                    $P^* \leftarrow P^* \setminus \{s\}$
                    $S \leftarrow S \cup N^-_G(s)$
            else
                // Pass the ball through
                if $s \in P_*$ then
                    $P_* \leftarrow P_* \setminus \{s\}$
                    $S \leftarrow S \cup N^+_G(s)$
    return $O \setminus V$
```

Appendix B. Bayes ball algorithm for identifying graph irrelevant nodes

The Bayes ball algorithm (Shachter, 1998) checks for D-separation in a graphical model and identifies graphically irrelevant nodes. For the sake of completeness, we present here the algorithm refined to the graphical model corresponding to the team form.

Given a team form $\mathcal{F}$, a target set T and an observation set O, we want to identify a subset $O' \subset O$ such that $O \setminus O'$ D-separates O' from T. (In Algorithm 2, T corresponds to dependent costs K_α and O corresponds to information set I_α.) Broadly speaking, the Bayes ball algorithm works by sending bouncing balls from each node in T into the network until they are absorbed in I. The ball passes through (from any parent to all children; from any child to all parents) the unobserved system dynamic and control nodes (i.e., nodes in $A \cup M \setminus O$). The ball bounces back (from any child to all children) from the unobserved primitive variable nodes (i.e., nodes in $B \setminus O$). For the observed nodes (i.e., nodes in O), the ball bounces back if it comes from the parents (to all parents) but is blocked if it comes from the children. Once all balls have been blocked, the set O' is the subset of O that has not been hit by a ball.

The precise algorithm is shown in Algorithm 7. We need to keep track of visited nodes to avoid repetitions. In the algorithm, V denotes the set of visited nodes, S the set of nodes that have received a ball but not passed or blocked it, R_* the set of nodes that have received a ball from their children, R^* the set of nodes that have received a ball from their parents, P_* the set of nodes that have *not* passed a ball to their children, and P^* the set of nodes that have *not* passed a ball to their parents. Note that P_* and P^* can overlap.

Appendix C. Proof of Proposition 3

Consider any policy $\mathbf{g} = (g_\alpha : \alpha \in A)$ of $(\mathcal{F}, \mathcal{T})$. Construct a policy $\mathbf{g}_H = (g^*_\alpha : \alpha \in A^*)$ of $(\mathcal{F}_H, \mathcal{T}_H)$ as follows: for all $\alpha_\circ \in A^*$,

$$g^*_{\alpha_\circ} = \begin{cases} g_{\alpha_\circ} & \alpha_\circ \in A \setminus H, \\ (\psi^\alpha_H : \alpha \in H), & \alpha_\circ = \lambda_H \end{cases}$$

where $\forall \alpha \in H$,

$$\psi^\alpha_H : \left(\mathbb{X}_{C_H}, \mathscr{F}_{C_H}\right) \to \left(\left(\mathbb{X}_{L_{H,\alpha}}, \mathscr{F}_{L_{H,\alpha}}\right) \to (\mathbb{X}_\alpha, \mathscr{F}_\alpha)\right)$$

is given by $\psi^\alpha_H(\cdot) = g_\alpha(\cdot, X_{C_H})$, where $g_\alpha(\cdot, X_{C_H})$ is a partial evaluation of g_α. By construction, $\mathbf{g}_H$ induces the same joint distribution on $\{X_n : n \in N^* \setminus \{\lambda_h\}\}$ in $(\mathcal{F}_H, \mathcal{T}_H)$ as $\mathbf{g}$ induces on $\{X_n : n \in N\}$ in $(\mathcal{F}, \mathcal{T})$. This same joint distribution and the identical cost structure of $\mathcal{F}$ and $\mathcal{F}_H$ imply that $\mathbf{g}$ and $\mathbf{g}_H$ yield the same expected cost.

Analogously, consider any policy $\mathbf{g}_H = (g^*_\alpha : \alpha \in A^*)$ of $(\mathcal{F}_H, \mathcal{T}_H)$. Construct a policy $\mathbf{g} = (g_\alpha : \alpha \in A)$ of $(\mathcal{F}, \mathcal{T})$ as follows: for all $\alpha_\circ \in A$,

$$g_{\alpha_\circ} = \begin{cases} g^*_{\alpha_\circ} & \alpha_\circ \in A \setminus H, \\ g^{\alpha_\circ}_{\lambda_H}(X_{C_H})(X_{L_{H,\alpha}}), & \alpha \in H \end{cases}$$

where $g^\alpha_{\lambda_H}$ is the α-component of g_{λ_H}. Again, by construction, $\mathbf{g}$ induces the same joint distribution on $\{X_n : n \in N\}$ in $(\mathcal{F}, \mathcal{T})$ as $\mathbf{g}_H$ induces on $\{X_n : n \in N^* \setminus \{\lambda_h\}\}$ in $(\mathcal{F}_H, \mathcal{T}_H)$. This same joint distribution and the identical cost structure of $\mathcal{F}$ and $\mathcal{F}_H$ imply that $\mathbf{g}_H$ and $\mathbf{g}$ yield the same expected cost.

References

Aicardi, M., Davoli, F., & Minciardi, R. (1987). Decentralized optimal control of Markov chains with a common past information set. *IEEE Transactions on Automatic Control*, *32*(11), 1028–1031.

Aumann, R. J. (1976). Agreeing to disagree. *Annals of Statistics*, *4*, 1236–1239.

Blackwell, D. (1964). Memoryless strategies in finite-stage dynamic programming. *Annals of Mathematical Statistics*, *35*(2), 863–865.

Gattami, A. (2007). *Generalized linear quadratic control*. (Ph.D. dissertation), Lund University.

Geiger, D., Verma, T., & Pearl, J. (1990). Identifying independence in Bayesian networks. *Networks*, *20*(5), 507–534.

Goldman, C. V., & Zilberstein, S. (2004). Decentralized control of cooperative systems. *Journal of Artificial Intelligence Research*, 143–174.

Grimmett, G. R., & Stirzaker, D. R. (2001). *Probability and random processes*. USA: Oxford University Press.

Ho, Y.-C. (1980). Team decision theory and information structures. *Proceedings of the IEEE*, *68*(6), 644–654.

Ho, Y.-C., & Chu, K.-C. (1972). Team decision theory and information structures in optimal control problems—part I. *IEEE Transactions on Automatic Control*, *17*(1), 15–22.

Mahajan, A., Martins, N., Rotkowitz, M., & Yüksel, S. (2012). Information structures in optimal decentralized control. In: *Proc. 51st IEEE conf. decision and control. Maui, Hawaii, December* (pp. 1291–1306).

Mahajan, A., & Tatikonda, S. (2009a). A graphical modeling approach to simplifying sequential teams. In: *Proc. 7th int. symp. modeling and optimization in mobile. ad hoc, and wireless networks, WiOpt* (pp. 1–8).

Mahajan, A., & Tatikonda, S. (2009b). Sequential team form and its simplification using graphical models. In: *Proc. 47th annual allerton conf. communication, control, and computing. Monticello, IL. September* (pp. 330–339).

Marschak, J., & Radner, R. (1972). *Economic theory of teams*. New Haven: Yale University Press.

Nayyar, A. (2011). *Sequential decision making in decentralized systems*. (Ph.D. dissertation), Ann Arbor, MI: University of Michigan.

Nayyar, A., Mahajan, A., & Teneketzis, D. (2013). Decentralized stochastic control with partial history sharing information structures: A common information approach. *IEEE Transactions on Automatic Control*, *58*(7), 1644–1658.

Radner, R. (1962). Team decision problems. *Annals of Mathmatical Statistics*, *33*, 857–881.

Sandell, N., Varaiya, P., Athans, M., & Safonov, M. (1978). Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, *23*(2), 108–128.

Shachter, R.D. (1998). Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In: *Proceedings of the fourteenth conference in uncertainty in artificial intelligence* (pp. 480–487).

Walrand, J. C., & Varaiya, P. (1983). Optimal causal coding-decoding problems. *IEEE Transactions on Information Theory*, *29*(6), 814–820.

Witsenhausen, H. S. (1968). A counterexample in stochastic optimum control. *SIAM Journal of Optimal Control*, *6*(1), 131–147.

Witsenhausen, H. S. (1971). On information structures, feedback and causality. *SIAM Journal of Control*, *9*(2), 149–160.

Witsenhausen, H. S. (1971). Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, *59*(11), 1557–1566.

Witsenhausen, H. S. (1975). The intrinsic model for discrete stochastic control: Some open problems. In *Control theory, numerical methods and computer system modelling*: *107* (pp. 322–335). Springer Verlag.

Witsenhausen, H. S. (1979). On the structure of real-time source coders. *Bell System Technical Journal*, *58*(6), 1437–1451.

Witsenhausen, H. S. (1988). Equivalent stochastic control problems. *Mathematics of Control Signals and Systems*, *1*, 3–11.

Yoshikawa, T. (1978). Decomposition of dynamic team decision problems. *IEEE Transactions on Automatic Control*, *23*(4), 627–632.

Aditya Mahajan received the B.Tech degree in Electrical Engineering from the Indian Institute of Technology, Kanpur, India, in 2003 and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from the University of Michigan, Ann Arbor, USA, in 2006 and 2008, respectively. He was a postdoctoral researcher at Uale University, New Haven from 2008 to 2010. In 2010, he joined the Electrical and Computer Engineering Department at McGill University, Montreal, QC, Canada, where he is an assistant professor. His principal research interests include decentralized stochastic control, team theory, multi-armed bandits, real-time communication, information theory, and discrete event systems.

Sekhar Tatikonda received the B.S., M.S., and Ph.D. (2000) in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA. He was a postdoctoral fellow at the University of California, Berkeley from 2000 to 2002. In 2002, he joined the Electrical Engineering Department at Yale University, New Haven, CT, where he is currently an associate professor. His research interests are in communication theory, information theory, stochastic control, distributed optimization, statistical machine learning and inference.