An axiomatic approach for simplification of sequential teams

Aditya Mahajan and Sekhar Tatikonda

February 15, 2011

Abstract

An axiomatic framework that identifies irrelevant data 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, and 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

1.1. Motivation

Multi-agent stochastic control systems in which all agents have a common objective are called teams [1, 2]. Such systems arise in almost all modern technologies including networked control systems, communication networks, sensor and surveillance networks, environmental remote sensing, smart grids, transportation networks, and robotics. In spite of these wide variety of applications, there is no general framework to address team decision problems. Most systems are addressed on a case-by-case basis. A common critical step in the solution framework is to identify if an agent can ignore part of the data available to it without any loss of performance. We refer to such a data as *irrelevant information*. Ignoring irrelevant information simplifies the search of an optimal control policy. As an example, consider the simplest stochastic control problem: the Markov decision process (MDP).

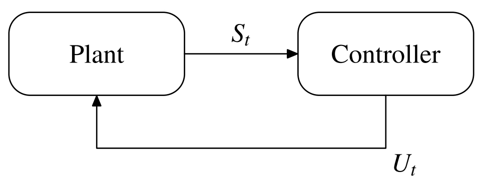


Figure 1: A control system representing a Markov decision process

Example 1 (Markov decision process) Consider the control system, shown in Figure 1, which consists of a plant and a controller. The state S_t of the plant evolves according to

$$S_{t+1} = f_t(S_t, U_t, W_t), \quad t = 1, 2, \dots, T-1$$

Submitted to IEEE Transactions on Automatic Control – February 15, 2011

where U_t is the control action of the controller and W_t is the plant disturbance at time t. The initial state S_1 of the plant is a random variable. The process $\{W_t, t = 1, 2, \dots, T\}$ is an independent process that is also independent of S_1. The controller observes the state S_t of the plant. It has perfect memory. So, it remembers everything that it has seen and done in the past and chooses a control action U_t based on the entire history, i.e.,

$$U_t = g_t(S_{1:t}, U_{1:t-1}),$$

where $S_{1:t}$ is a shorthand notation for $(S_1, \dots, S_t)$ and similar interpretation holds for $U_{1:t-1}$. At each time a cost $c_t(S_t, U_t)$ is incurred. The control objective is to choose a *control policy* $\mathbf{g} := (g_1, g_2, \dots, g_T)$ to minimize the expected total cost

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

□

At first glance, the above problem appears daunting because the control action U_t may depend on the entire history $(S_{1:t}, U_{1:t-1})$. However, a standard result in Markov decision theory [3] shows that the *past data* $(S_{1:t-1}, U_{1:t-1})$ is irrelevant. In other words, without loss of optimality, the controller may choose action U_t based only on he current state S_t, i.e.,

$$U_t = g_t(S_t) \tag{1}$$

This *structural result* is then used to derive a dynamic programming decomposition to find the control law of the form (1) (see a standard textbook on Markov decision theory like [3] for details).

The above structural result is robust to various modeling assumptions: the specifics of the state spaces, the underlying probability measure, and the specifics of the plant dynamics and cost functions. All that matters is the *form* of the system.

In this paper we present an algorithm to identify irrelevant information available at a control agent in a sequential teams. For that matter, we define a notion of *sequential team form* that captures the properties of sequential teams that depend only its structure and not on the specifics of the state spaces, the probability measures, the plant dynamics, and the cost functions. We model the sequential team form as a directed graph and present an algorithm that identifies the edges of the graph that may be removed without loss of performance.

1.2. Examples

Besides the Markov decision process, we use two examples to illustrate the concepts developed in this paper. Below, we describe these examples.

The first example is a real-time communication system originally studied by Witsenhausen in [4]

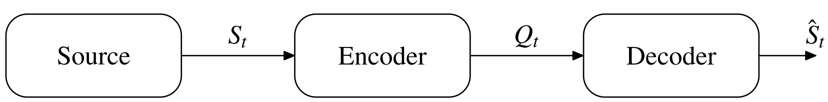


Figure 2: A real-time communication system

Example 2 (Real-time communication) Consider a real-time communication system, shown in Figure 2, which consists of a source, an encoder, and a decoder. The source is a first-order

Markov process $\{S_t, t = 1, \dots, T\}$. The encoder observes the source output and generates quantized symbols Q_t, causally and in real-time, as follows

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

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

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

□

For this example, Witsenhausen [4] showed that we can restrict attention to encoders of the form

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

without loss of optimality. Thus, M_{t-1} is *indirectly observed* at the encoder, and $(S_{1:t-1}, Q_{1:t-1})$ is *irrelevant*.

The next example is a decentralized control system where the state and control actions of the system is observed by all control stations after one-step delay. It is a special case of a general model of *delayed state sharing information structure*, considered in [5], which assumes an arbitrary delay k in the sharing of states and control actions. It is also a special case of a general model of *delayed observation sharing information structure*, considered in [6–8], which assumes noisy observations at the controllers and arbitrary delay k in the sharing of observations and control actions. Both these models consider a general system with N control stations. For ease of exposition, we consider a system with 2 control stations.

Example 3 (Decentralized conrtol with one-step delayed sharing of state) Consider a decentralized control system, shown in Figure 3, which consists of two coupled subsystems. The state S^i_t of subsystem i, $i = 1, 2$, evolves according to

$$S^i_{t+1} = f^i_t(S^1_t, S^2_t, U^1_t, U^2_t, W^i_t)$$

where U^j_t is the control action of controller j, $j = 1, 2$ at time t and W^i_t is the plant disturbance of subsystem i. The initial states S^1_t and S^2_t of the subsystems are independent random variables. The processes $\{W^1_t, t = 1, 2, \dots, T\}$ and $\{W^2_t, t = 1, 2, \dots, T\}$ are independent processes that are also independent from each other and from S^1_1 and S^2_1. Controller i observes the state S^i_t of the subsystem i and the one step delayed state and control action (S^j_{t-1}, U^i_{t-1}) of subsystem j, $j \neq i$. Both controllers have perfect memory. So, control actions (U^1_t, U^2_t) are chosen as

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

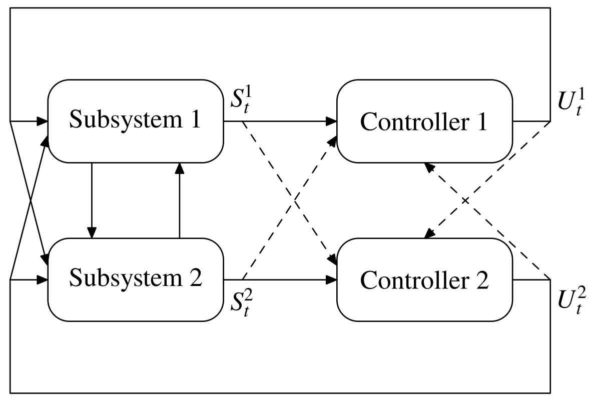


Figure 3: Coupled subsystems with decentralized controllers. The dashed lines indicate links with one-unit delay.

At each time a cost $c_t(S_t^1, S_t^2, U_t^1, U_t^2)$ is incurred. The control objective is to choose control policies $\mathbf{g}^1 := (g_1^1, \dots, g_T^2)$ and $\mathbf{g}^2 := (g_1^2, \dots, g_T^2)$ to minimize the expected cost

$$\mathbb{E}_{(\mathbf{g}^1, \mathbf{g}^2)}\Big[\sum_{t=1}^{T} c_t(S_t^1, S_t^2, U_t^1, U_t^2)\Big] \qquad \square$$

For the above model of one-step one-step delayed sharing of state, the results of [5] as well as the results of [6–8] simplify to the following: without loss of optimality, we can restrict attention to control laws of the form

$$\begin{aligned} U_t^1 &= g_t^1(S_t^1, S_{t-1}^1, S_{t-1}^2, U_{t-1}^1, U_{t-1}^2), \quad t = 2, 3, \dots \\ U_t^2 &= g_t^2(S_{t-1}^1, S_t^2, S_{t-1}^2, U_{t-1}^1, U_{t-1}^2). \quad t = 2, 3, \dots \end{aligned}$$

This results shows that the two-step delayed past states and control actions $(S_{1:t-2}^1, S_{1:t-2}^2, U_{1:t-2}^1, U_{1:t-2}^2)$ are *irrelevant* for control.

The results of the above three examples are similar in nature. In all the three cases, part of the data observed at the agents is irrelevant and can be ignored by the agents without loss of optimality. However, the results are proved using conceptually different approaches. The structural result for MDPs is proved by simultaneously deriving a dynamic programming decomposition [3]. The structural result for real-time communication is proved by first establishing the structural result for two and three stage systems, and then extending them for arbitrary time by aggregating time to artificially construct a three stage system and using backward induction [4]. The structural result for the one-step delayed state sharing is proved by constructing a controlled Markov chain for the decentralized system [5, 7]. Thus, the solution approach in each case is problem dependent. However, the critical step in all the proofs relies, directly or indirectly, on Blackwell's principle of irrelevant information [9, 10].

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 $c : \mathbb{X} \times \mathbb{U} \mapsto \mathbb{R}$ be a bounded Borel-measurable function. Then for any Borel-measurable function $g : \mathbb{X} \times \mathbb{Y} \mapsto \mathbb{U}$, there exists another Borel measurable function $h : \mathbb{X} \mapsto \mathbb{U}$ such that*

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

where the expectation is with respect to P. □

In this paper, we separate out this critical step from rest of the algebraic book-keeping. We then present a graphical-models based approach for recursively removing irrelevant information from all agents in the system.

1.3. Contributions

The main contributions of this paper are:

1. We present a framework for modeling a sequential team as a directed graph or a Bayesian network. This framework is general enough to model any unconstrained sequential team problem. In spite of its generality, we believe that, pedagogically, this is an easier model to understand the derivation of the structural results in sequential teams.

2. We present an axiomatic approach to derive structural results for sequential teams. This methodology does not require us to derive a dynamic programming decomposition to derive structural results. Such a decoupled derivation is very useful for decentralized control systems where a dynamic programming decomposition is usually difficult to obtain.

3. The methodology for deriving structural results uses standard algorithms from graphical models and, as such, is easy to automate. This allows us to write a computer program to derive structural results for sequential teams.

1.4. 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-field $\{\mathscr{F}_n\}$, $\mathscr{F}_N$ is a short-hand for $\bigotimes_{n \in N} \mathscr{F}_n$.

2. Modeling

We begin by an informal description of a sequential team and follow that by a formal definition.

2.1. Sequential teams

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 evolves with time and the *system variables* $\{S_m : m \in M\}$ are generated. 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 α, $\alpha \in A$, 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 (which is also denoted by U_α), according to a *control law*

$$g_\alpha : \Big(\prod_{n \in I_\alpha} \mathbb{X}_n, \bigotimes_{n \in I_\alpha} \mathscr{F}_n \Big) \mapsto (\mathbb{X}_\alpha, \mathscr{F}_\alpha)$$

as

$$X_\alpha = g_\alpha(X_n : n \in I_\alpha)$$

The collection $\mathbf{g} \coloneqq (g_\alpha : \alpha \in A)$ is called a *control policy.*

Each system variable X_m (which is also denoted by S_m), $m \in M$, is also associated with an information set I_m and a *dynamics function*

$$f_m : \Big(\prod_{n \in I_m} \mathbb{X}_n, \bigotimes_{n \in I_m} \mathscr{F}_n \Big) \mapsto (\mathbb{X}_m, \mathscr{F}_m)$$

The variable X_m is generated as

$$X_m = f_m(X_n : n \in I_m)$$

The collection $(f_m : m \in M)$ is called the *system dynamics.*

The system is *sequential.* This means that a bijection

$$\varphi : N \mapsto \{1, 2, \dots, |N|\}$$

exists such that for any $n \in \varphi(A \cup M)$

$$I_{\varphi^{-1}(n)} \subseteq \varphi^{-1}(\{1, 2, \dots, n-1\})$$

See [11] for details. 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 [11], 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

$$\begin{aligned} P_{\mathbf{g}}(X_n : n \in N) = & \bigotimes_{\beta \in B} Q(X_\beta) \\ & \otimes \bigotimes_{m \in M} \mathbb{1}[X_m = f_m(X_n : n \in I_m)] \\ & \otimes \bigotimes_{\alpha \in A} \mathbb{1}[X_\alpha = g_\alpha(X_n : n \in I_\alpha)] \end{aligned} \tag{2}$$

The performance of a control policy is quantified by expectation (with respect to $P_{\mathbf{g}}$) of a cost function

$$\sum_{k \in K} c_k(X_n : n \in D_k) \tag{3}$$

where K is a finite set, the sets D_k, $k \in K$, are subsets of N and

$$c_k : \Big(\prod_{n \in D_k} \mathbb{X}_n, \bigotimes_{n \in D_k} \mathscr{F}_n \Big) \mapsto (\mathbb{R}, \mathscr{B})$$

where $\mathscr{B}$ is the Borel σ-algebra over reals.

Now we present a formal definition of a sequential team.

Definition 1 (Sequential Team) A sequential team is a tuple that consists of seven 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. *Measurable Spaces* $\{(\mathbb{X}_n, \mathscr{F}_n) : n \in N\}$:
   - Variable X_n, $n \in N$ takes value in measurable space $(\mathbb{X}_n, \mathscr{F}_n)$.
3. *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 $A \cup M$.
4. *Probability Measure Q*
   - Q is a product measure on the primitive random variables $\{X_\beta : \beta \in B\}$
5. *System dynamics* $\{f_m : m \in M\}$:

- f_m is a measurable function

$$f_m : \Big(\prod_{n \in I_m} \mathbb{X}_n, \bigotimes_{n \in I_m} \mathscr{F}_n \Big) \mapsto (\mathbb{X}_m, \mathscr{F}_m)$$

  such that

$$X_m = f_m(X_n : n \in I_m)$$

6. *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 k-th cost term.
7. *Cost functions* $\{c_k : k \in K\}$
   - c_k denotes the k-th cost function and

$$c_k : \Big(\prod_{n \in D_k} \mathbb{X}_n, \bigotimes_{n \in D_k} \mathscr{F}_n \Big) \mapsto (\mathbb{R}, \mathscr{B})$$

   - The total cost incurred in the system is

$$\sum_{k \in K} c_k(X_n : n \in D_k) \qquad \square$$

Definition 2 (Control Policy) A control policy for a sequential team is a collection $\mathbf{g} \coloneqq \{g_\alpha : \alpha \in A\}$ such that

$$g_\alpha : \Big(\prod_{n \in I_\alpha} \mathbb{X}_n, \bigotimes_{n \in I_\alpha} \mathscr{F}_n \Big) \mapsto (\mathbb{X}_\alpha, \mathscr{F}_\alpha).$$

Given a control policy, the control variables are generated as

$$X_\alpha = g_\alpha(X_n : n \in I_\alpha) \qquad \square$$

2.2. Some technical remarks

1. Traditionally, instead of using primitive random variables, the system dynamics are described by a Markov kernel on the next state conditioned on the current state and the control action. Both these representations are equivalent.
2. There is no loss of generality in assuming that the primitive random variables are independent.
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 [12], which, in turn, was shown to be equivalent to the intrinsic model [11, 13] 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.

2.3. Team form and team type

Usually, one is interested in the following stochastic control problem.

Problem 1 (Optimal control and value of a sequential team) *Given a sequential team, choose a control policy* $\mathbf{g}$ *to minimize*

$$\mathbb{E}_{\mathbf{g}}\Big[\sum_{k \in K} c_k(X_n; n \in D_k)\Big]$$

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.*

However, we are interested in a related but different question. To explain that question, we need two definitions.

Definition 3 (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

$$g_\alpha : \Big(\prod_{n \in J_\alpha} \mathbb{X}_n, \bigotimes_{n \in J_\alpha} \mathscr{F}_n\Big) \mapsto (\mathbb{X}_\alpha, \mathscr{F}_\alpha)$$

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. □

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*

To capture sufficient and irrelevant information, we split the specification of a sequential team into two parts—team form and team type.

Definition 4 (Team Form) A *team form* is specified by three components

1. Variable structure (N, A, B, M)
2. Information structure $\{I_n : n \in A \cup M\}$
3. Cost structure $(K, \{D_k : k \in K\})$

where each component satisfies the corresponding properties of Definition 1. □

Definition 5 (Team Type) A *team type* type is specified by four components

1. Measurable spaces $\{(\mathbb{X}_n . \mathscr{F}_n) : n \in N\}$
2. Probability measure Q
3. System dynamics $\{f_m : m \in M\}$
4. Cost functions $\{c_k : k \in K\}$

where each component satisfies the corresponding properties of Definition 1. □

The team forms of the three examples described in the introduction are as follows:

Example 1 (Markov decision process, continued) A Markov decision process is a sequential team with $|N| = 3T$ variables—T process noise variables, T state variables, and T control variables.

For ease of notation, instead of specifying the variable structure by the indices (N, A, B, M), we specify it by the sets (X_N, X_A, X_B, X_M). The components of the team form are:

1. *Variable structure*
   - Control variables $X_A = \{U_1, U_2, \dots, U_T\}$.
   - Primitive variables $X_B = \{S_1, W_1, W_2, \dots, W_T\}$.
   - System variables $X_M = \{S_2, S_3, \dots, S_T\}$.
   - All variables $X_N = X_A \cup X_B \cup X_M$.
2. *Information structure*
   - For $X_n = U_t \in X_A$, $t = 1, \dots, T$, $I_n = \{S_1, S_2, \dots, S_t, U_1, U_2, \dots, U_{t-1}\}$.
   - For $X_n = S_t \in X_M$, $t = 1, \dots, T$, $I_n = \{S_{t-1}, W_{t-1}, U_{t-1}\}$.
3. *Cost structure*
   - Number of cost terms $K = T$.
   - For $t = 1, \dots, T$, $D_t = \{S_t, U_t\}$.

The results of Markov decision theory show that $\{X_t\}$ (respectively, $\{X_{1:t-1}, U_{1:t-1}\}$) is sufficient (respectively, irrelevant) information for control at the control station at time t.

Example 2 (Real-time communication, continued) We can always represent a first-order Markov process $\{S_1, \dots, S_T\}$ as

$$S_t = \phi_t(S_{t-1}, W_{t-1})$$

using the inverse transform method or Smirnov transform [14, Sec 4.11] such that $\{W_1, \dots, W_T\}$ is an independent process that is also independent of the initial state S_1. The process $\{W_t, \dots, W_T\}$ is called the source noise process. With this representation, the real-time communication problem of Example 2 is a sequential team with $N = 5T$ variables— T source noise variables, T source output variables, T quantized symbol variables, and T receiver memory variables, and T source estimation variables.

As before, we specify the variable structure by the sets (X_N, X_A, X_B, X_M). The components of the team form are:

1. *Variable structure*
   - Control variables: $X_A = \{Q_1, Q_2, \dots, Q_T, M_1, M_2, \dots, M_{T-1}, \hat{S}_1, \hat{S}_2, \dots, \hat{S}_T\}$
   - Primitive variables: $X_B = \{S_1, M_0, W_1, W_2, \dots, W_T\}$
   - System variables: $X_M = \{S_2, S_3, \dots, S_T\}$
   - All variables: $X_N = X_A \cup X_B \cup X_M$
2. *Information structure*
   - For $X_n = Q_t \in X_A$, $t = 1, 2, \dots, T$, the information set $I_n = \{S_1, S_2, \dots, S_t, Q_1, Q_2, \dots, Q_{t-1}\}$
   - For $X_n = M_t \in X_A$, $t = 1, 2, \dots, T-1$, the information set $I_n = \{Q_t, M_{t-1}\}$
   - For $X_n = \hat{S}_t \in X_A$, $t = 1, 2, \dots, T$, the information set $I_n = \{S_{t-1}, W_{t-1}\}$.

3. *Cost structure*
   - Number of cost terms $K = T$
   - For $t = 1, 2, \dots, T$, $D_t = \{S_t, \hat{S}_t\}$

The results of [4] show that $\{S_t, M_{t-1}\}$ (respectively, $\{S_{1:t-1}, Q_{1:t-1}, M_{1:t-1}\}$) is sufficient (respectively irrelevant) for control at the encoder at time t.[1]

Example 3 (Decentralized control with one-step delayed sharing of state, continued) The decentralized control system of Example 3 is a sequential team with $N = 6T$ variables—$2T$ state variables, $2T$ control variables, and $2T$ process noise variables.

As before, we specify the variable structure by the sets (X_N, X_A, X_B, X_M). The components of the team form are:

1. *Variable structure*
   - Control variables $X_A = \{U^1_1, U^1_2, \dots U^1_T, U^2_1, U^2_2, \dots, U^2_T\}$.
   - Primitive variables $X_B = \{S^1_1, S^2_1, W^1_1, W^1_2, \dots, W^1_T, W^2_1, W^2_2, \dots, W^2_T\}$
   - System variables $X_M = \{S^1_2, S^1_3, \dots, S^1_T, W^2_2, S^2_3, \dots, S^2_T\}$.
   - All variables: $X_N = X_A \cup X_B \cup X_M$
2. *Information structure*
   - For $X_n = U^i_t \in X_A$, $i = 1, 2$. For $t = 1, \dots, k-1$, $I_n = \{S^i_1, S^i_2, \dots, S^i_t, U^i_1, U^i_2, \dots, U^i_{t-1}\}$; for $t = k, k{+}1, \dots, T$, $I_n = \{S^i_1, S^i_2, \dots, S^i_t, S^j_1, S^j_2, \dots, S^j_t, U^i_1, U^i_2, \dots, U^i_{t-1}, U^j_1, U^j_2, \dots, U^j_{t-k-1}\}$, where $j \in \{1, 2\}$, $j \neq i$.
   - For $X_n = S^i_t \in X_M$, $i = 1, 2$, $i = 2, \dots, T$, $I_n = \{S^1_{t-1}, S^2_{t-1}, U^1_{t-1}, U^2_{t-1}\}$.
3. *Cost structure*
   - Number of cost terms $K = T$.
   - For $t = 1, \dots, T$, $D_t = \{S^1_t, S^2_t, U^1_t, U^2_t\}$.

The result of [5] show that $\{S^i_{t-k:t}, S^j_{t-k}, U^i_{t-k:t-k-1}\}$ (respectively, $\{S^1_{1:t-k-1}, S^2_{1:t-k-1}, U^1_{1:t-k-1}, U^2_{1:t-k-1}\}$) is sufficient (respectively, irrelevant) for control at station i at time t.

2.4. Graphical representation of a team form

We first explain the terminology of graphical models, then present the directed graphs corresponding to Examples 1 and 2, and finally show how to obtain a directed graph for a general team form.

2.4.1. Directed graphs

A directed graph G is a tuple (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*

[1] Although, the data $\{M_1, \dots, M_{t-1}\}$ is not part of the information set of the encoder at time t, it is a function of the information set. Therefore, we include $\{M_1, \dots, M_{t-1}\}$ in the specification of sufficient and irrelevant data for control. See Section 4.1 on strict expansion of information structures for details.

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* of *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

$$\overrightarrow{U} = \bigcup_{u \in U} \overrightarrow{u}$$

2.4.2. Examples

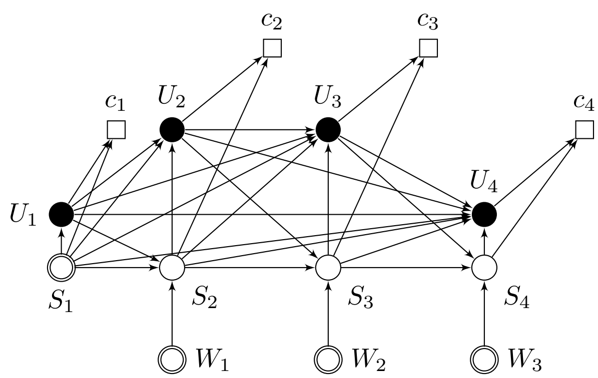


Figure 4: A directed acyclic graph corresponding to the team form of a Markov decision process (Example 1)

Example 1 (Markov decision process, continued) A directed graph for the team form of a Markov decision process for horizon $T = 4$ is shown in Figure 4. This graph is a directed acyclic graph with four types of vertices: filled circles ● representing control variables, double circles ◎ representing primitive variables, empty circles ○ representing system variables and empty squares □ indicating cost terms. For the vertices corresponding to control and system variables, the in-neighborhood corresponds to the information set of the variable. For example, the in-neighborhood of U_3 in Figure 4 is $\{S_1, S_2, S_3, U_1, U_2\}$—the same as the information set of U_3. The vertices corresponding to primitive random variables do not have any incoming edges. The in-neighborhood of the vertices corresponding to a cost term corresponds to the variables coupled due to that cost term. For example, the in-neighborhood of c_3 in Figure 4 is $\{S_3, U_3\}$, which are the arguments of the cost function c_3.

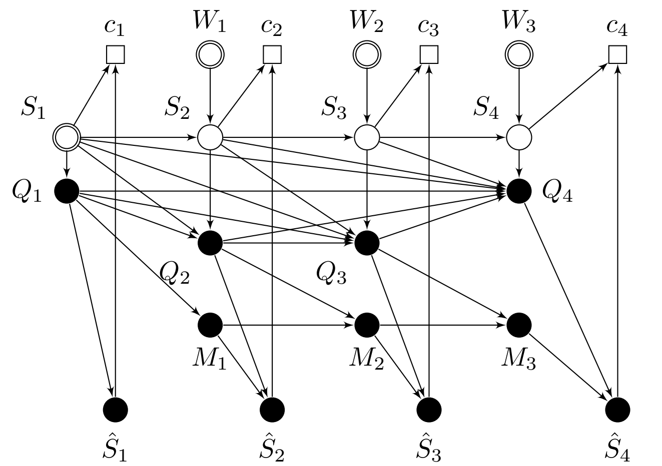


Figure 5: A directed acyclic graph corresponding to the team form of real-time communication (Example 2)

Example 2 (Real time communication, continued) A directed graph for the team form of real-time communication for horizon $T = 4$ is shown in Figure 5. As before, the graph has four types of vertices: full circles ● representing control variables, double circles ⊚ representing primitive random variables, empty circles ○ representing system variables, and empty squares □ indicating cost terms.

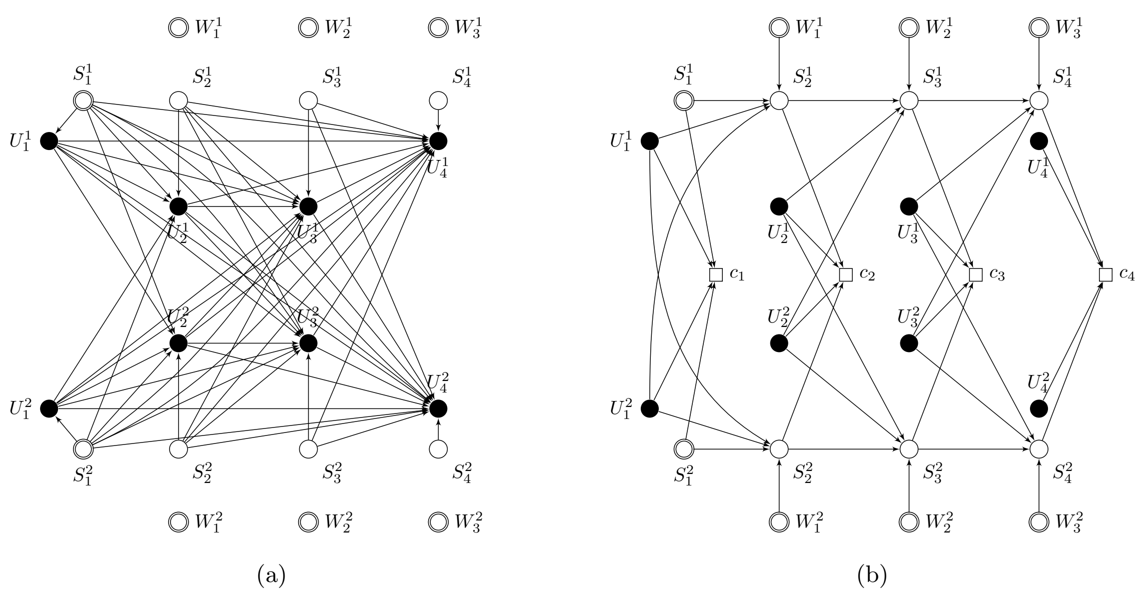


Figure 6: A directed acyclic graph corresponding to the team form of decentralized control system of Example 3. For clarity, we have split the edges into two graphs. The first graph shows the information sets of the control variables while the second graph shows the information sets of the system variables and the cost structure.

Example 3 (Decentralized control with one-step delayed sharing of state, continued) A directed graph for the team form of decentralized control system for horizon $T = 4$ and delay $k = 2$ is shown in Figure 6. As before, the graph has four types of vertices: full circles ● representing control variables, double circles ⊚ representing primitive random variables, empty circles ○ representing system variables, and empty squares □ indicating cost terms. As this graph is a bit complicated, we have split the edges in two parts for clarity. The first graph shows the information sets of the control variables, while the second graph shows the information sets of the system variables and the cost structure.

2.4.3. Modeling a team form as a DAG

A general 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 between X_m and X_n.
  2. For all $k \in K$ and $n \in D_k$ draw an edge between X_n and c_k

  Thus, the edge set E of the DAG is

$$E = \Big(\bigcup_{n \in A \cup M} \bigcup_{m \in I_n} (X_m, X_n) \Big) \cup \Big(\bigcup_{k \in K} \bigcup_{n \in D_k} (X_n, c_k) \Big)$$

2.4.4. 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 (2). This factorization is of the form

$$\mathbb{P}(X_n : n \in N) = \bigotimes_{n \in N} \mathbb{P}(X_n | X_m : m \in N_G^-(n))$$

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* 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 [15]. For completeness, we summarize the basic results of Bayesian networks in the next section. See [15, 16] for details.

2.5. Bayesian networks with deterministic nodes

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

$$\mathbb{P}(X_v : v \in V) = \bigotimes_{v \in V} \mathbb{P}(X_v | X_n : n \in N_G^-(v))$$

Definition 6 (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)

$$\mathbb{P}(X_n : n \in S_1 | X_n : n \in S_2 \cup S_3) = \mathbb{P}(X_n : n \in S_1 | X_n : n \in S_2), \quad \mathbb{P} - \text{a.s.} \qquad \square$$

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) 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

Figure 7: Head-to-head and tail-to-tail vertices in a trail.

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. 7(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. 7(b). A vertex that starts of end 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 7 (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. $\square$

Definition 8 (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-separate* S_1 from S_3 iff there exist no trails t between a vertex in S_1 and a vertex in S_3 along which

1. Every head-to-head vertex is either in S_2 or has a descendant in S_2
2. No other vertex is functionally determined by S_2 $\square$

A standard result in Bayesian networks is the following [15, 16].

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.* $\square$

D-separation can be checked in $O(|E|)$ time by either using an edge labeling variation of breadth first search [15] or using an edge based reachability analysis (called Bayes Ball) [16].

2.6. Other ways of modeling sequential teams as a DAG

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, but these approaches have not gained much popularity.

The first approach is by Witsenhausen [4, 6] who modeled a sequential team as a DAG in which a node represents either a system dynamics function or a control law; a directed edge between nodes f and g means that the output of f is an input of g. Thus, this graph captures the input-output relationship between the variables. For example, according to this approach, the real-time communication system of Example 2 would be represented as a DAG shown in Figure 8.

Another approach was proposed by Ho and Chu [17] and later refined by Yoshikawa [18]. They modeled a sequential team as a DAG in which a node represents an agent. There are two types of directed edges: a solid edge between nodes u and v implies that agent u must act before agent v; a dashed edge between u and v implies that agent v knows all the information available to agent u, that is, $I_u \subseteq I_v$. Thus, this graph captures the precedence relationship and the nestedness of information between the agents. For example, according to this approach, the real-time communication system of Example 2 would be represented as a DAG shown in Figure 9.

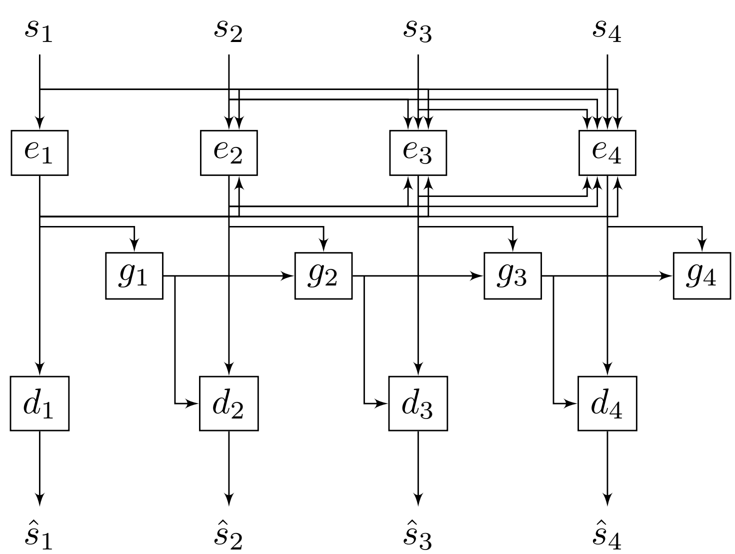


Figure 8: The real-time communication system of Example 2 as a DAG according to a model proposed by Witsenhausen [4, 6].

Both these models fail to capture the conditional independence relationship between the system variables—the key property that allows us to identify structural results using a DAG. A model similar to the one presented in this paper was used by [19] to model centralized stochastic control systems.

3. Problem Formulation

In this section, we formally pose the question of identifying structural properties of sequential teams.

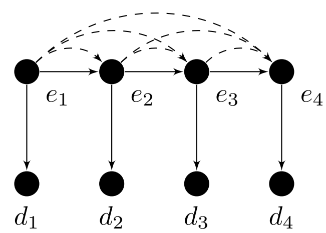


Figure 9: The real-time communication system of Example 2 as a DAG according to a model proposed by Ho and Chu [17] and Yoshikawa [18].

Definition 9 (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 $\mathcal{F}' = \langle (N', A', B', M'), \{I'_n : n \in A' \cup M'\}, (K' : \{D'_k, k \in K'\})$ are *equivalent* if

1. They have the same variable and cost structures, i.e.,
$$(N, A, B, M) = (N', A', B', M'), \quad K = K', \quad \text{and } \forall k \in K, D_k = D'_k$$
2. The information sets of the system dynamics are identical, i.e.,
$$\forall m \in M, I_m = I'_m$$
3. The *any choice* of team types $\mathcal{T}$, the values of the teams $(\mathcal{F}, \mathcal{T})$ and $(\mathcal{F}', \mathcal{T})$ are the same. $\square$

Definition 10 (Simplification of a team form) A team form $\mathcal{F}' = \langle (N', A', B', M'), \{I'_n : n \in A' \cup M'\}, (K' : \{D'_k, k \in K'\})$ 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 on the of inclusions is strict. $\square$

Definition 11 (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 $\mathcal{F}'$. $\square$

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 simplification.*

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 $\mathcal{F}$

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.

4. Simplification at a single agent

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

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

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

4.1. Strict expansion of an information structure

The idea of a strict expansion of an information structure was introduced in [11]. 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 12 (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 $\alpha \in A$

$$I_n \subset I_\alpha \implies X_n \in I_\alpha$$

□

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

The information structure of Examples 1 and 3 are strictly expanded, while that of Example 2 is not. It can be stricitly expanded using Algorithm 1. However, the strict expansion of a team form is not unique. Depending on the order in which we proceed, we may end up with different strict expansions. Two of the possible strict expansions of the information structure of Example 2 are showin in Figure 10.

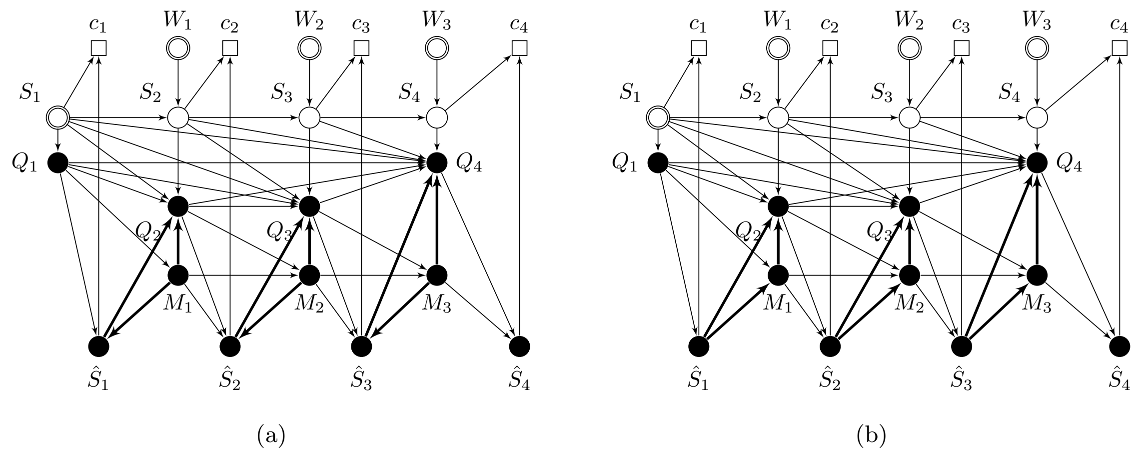


Figure 10: Two possible strict expansions of the information structure of the team form of Example 2. The thick lines denotes the edges added as part of the expanding the information structure.

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 $\mathcal{F}$ result in a team form that is equivalent to $\mathcal{F}$.* □

This is proved in Appendix A.

4.2. Main result for single agent

Before describing the main result, we need one more definition.

Definition 13 (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.* □

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

$$C(K') = \sum_{k \in K'} c(X_n; n \in D_k)$$

The total cost can be decomposed as

$$C(K) = C(K_\alpha) + C(K \setminus K_\alpha)$$

The definition of dependent cost K_α implies that $C(K \setminus K_\alpha)$ does not depend on X_α and hence on g_α. Thus,

$$\arg\min_{g_\alpha} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K)] = \arg\min_{g_\alpha} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha)] \tag{4}$$

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

$$\mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha) | X_{I_\alpha}, X_\alpha] = \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha) | X_{S_\alpha}, X_\alpha] \tag{5}$$

Now, we can combine (4) and (5) with Blackwell's principle of irrelevant information (Theorem 1) to complete the proof as follows: Let

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

(4) implies that agent α is interested in minimizing

$$\begin{aligned}
\mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha)] &= \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}\big[\mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[C(K_\alpha) | X_{I_\alpha}, X_\alpha]\big] \\
&\overset{(a)}{=} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}\big[\mathbb{E}_{\mathbf{g}_{-\alpha}}[C(K_\alpha) | X_{I_\alpha}, X_\alpha]\big] \\
&\overset{(b)}{=} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}\big[\mathbb{E}_{\mathbf{g}_{-\alpha}}[C(K_\alpha) | X_{S_\alpha}, X_\alpha]\big] \\
&\overset{(c)}{=} \mathbb{E}_{(\mathbf{g}_{-\alpha}, g_\alpha)}[h(X_\alpha, X_{S_\alpha})] \\
&= \mathbb{E}_{\mathbf{g}_{-\alpha}} h(g_\alpha(X_{S_\alpha}, X_{J_\alpha}), X_{S_\alpha})]
\end{aligned} \tag{6}$$

where (a) follows from policy independence of conditional expectation [11], (b) follows from (5), and (c) follows from the definition of $h(\cdot)$. Now, Blackwell's principle of irrelevant information (Theorem 1) implies that to minimize the RHS of (6), there is no loss of optimality in choosing X_α as

$$X_\alpha = g_\alpha(X_n : n \in S_\alpha)$$

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 $(I_\alpha \setminus J_\alpha, \alpha)$ 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. 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 [16]. This algorithm, refined for the graphical model corresponding to the team form, is presented in Algorithm 7 in Appendix B. After identifying graphically irrelevant data J_α using Algorithm 7, we can simplify a team form at agent α using Algorithm 2. Since this algorithm is an implementation of Theorem 2, the output team form is a simplification of the input 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

Algorithm 2: SimplifySingleAt

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 α
output: Team form $\mathcal{F}'$
output: Irrelevant data J_α
begin
  let $K_\alpha \coloneqq \{k \in K : D_k \cap \overrightarrow{X}_\alpha \neq \emptyset\}$
  let $J_\alpha \coloneqq$ `GraphIrrelevant(`$\mathcal{F}, I_\alpha \cup \{\alpha\}, K_\alpha$`)`
  $I_\alpha \leftarrow I_\alpha \setminus J_\alpha$
  return $(\mathcal{F}, J_\alpha)$

Algorithm 3: SimplifyTeam (First 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 $\mathcal{F}' \coloneqq \mathcal{F}$
    forall $\alpha \in A$ **do**
      $(\mathcal{F}, J) \leftarrow$ `SimplifySingleAt(`$\mathcal{F}, \alpha$`)`
  until $\mathcal{F}' = \mathcal{F}$

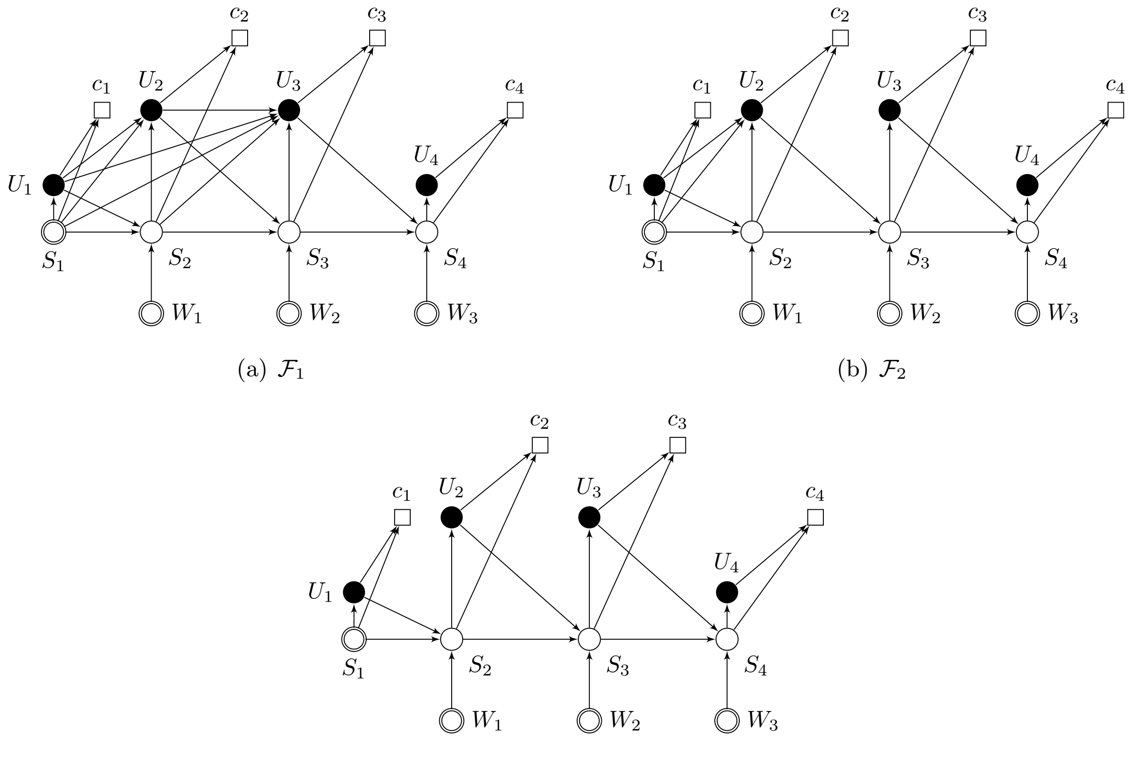
each iteration simplifies the original team form, the final team form is also a simplification of the input team form.

As we show in the next section, we can algorithmically rederive the structural results of Examples 1 and 2 using Algorithm 3. However, Algorithm 3 does not help in simplifying the model of Example 3. For that matter, we need to consider a coordinator of a group of agents, which is described in Section 5.

4.3. Examples

Example 1 (Markov decision process, continued) Consider the team form of a Markov decision process shown in Figure 4. Denote this team form by $\mathcal{F}_0$. The information structure of this team form is strictly expanded. Denote the dependent costs by $K_t = K_{U_t}$, $t = 1, \dots, 4$. These dependent costs are given by: $K_1 = \{c_1, c_2, c_3, c_4\}$, $K_2 = \{c_2, c_3, c_4\}$, $K_3 = \{c_3, c_4\}$, and $K_4 = \{c_4\}$. Algorithm 3 on this team form proceeds as follows. We order the agents as $A = \{U_4, U_3, U_2, U_1\}$.[2] First, we need to obtain $\mathcal{F}_1 =$ `SimplifySingleAt(`$\mathcal{F}_0, U_4$`)`. For that matter, we need to find $J_4 =$ `GraphIrrelevant(`$\mathcal{F}_0, \{S_{1:4}, U_{1:4}\}, \{c_4\}$`)` using Algorithm 7. In Algorithm 7, balls are passed[3] from c_4 to its parents $\{S_4, U_4\}$; since both these nodes are part of the observed data $\{S_{1:4}, U_{1:4}\}$, both of them block the ball and the algorithm terminates. Thus, the irrelevant nodes are $J_4 = \{S_{1:3}, U_{1:3}\}$ which are not visited. Removing the edges from J_4 to U_4 gives $\mathcal{F}_1$, which is shown in Figure 11(a). Proceeding along these lines, simplifying $\mathcal{F}_1$ at U_3 gives $\mathcal{F}_2$ shown in Figure 11(b). Simplifying $\mathcal{F}_2$ at U_2 gives $\mathcal{F}_3$ shown in Figure 11(c). $\mathcal{F}_3$ does not change after Algorithm 3 iterates through it

[2]Any other ordering will work in the same manner with a few extra steps where the team form does not change.
[3]See Appendix B for details.



(a) $\mathcal{F}_1$

(b) $\mathcal{F}_2$

(c) $\mathcal{F}_3$

Figure 11: Different steps in the simplification of the team form of Figure 1. $\mathcal{F}_1$ is obtained by simplifying the team form of Figure 1 at U_4; $\mathcal{F}_2$ by simplifying $\mathcal{F}_1$ at U_3 and $\mathcal{F}_3$ by simplifying $\mathcal{F}_2$ at U_2.

one more time. Hence, $\mathcal{F}_3$, shown in Figure 11(c) is a simplification of the team form of Figure 4 obtained by Algorithm 3.

The graphical model of Figure 11(c) corresponds to a team form in which the informations sets at U_t are $\{S_t\}$. Thus, we have recovered the structural result of MDPs using Algorithm 3.

Example 2 (Real time communication, continued) Consider the team form of a real-time communication system shown in Figure 5. The first step of Algorithm 3 is to strictly expand its information structure. Depending on the order in which we proceed, we can get different expansions. Two of them are shown in Figure 10. In this case, irrespective of which expansion we choose, after simplifying the team form at all control variables one by one we get the team form shown in Figure 12.

The graphical model of Figure 12 corresponds to a team form where 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 $\{Q_t, M_{t-1}\}$.

Example 3 (Decentralized control with one-step delayed sharing of state, continued) Consider the team form of a decentralized control system shown in Figure 6. Algorithm 3 does not simplify this team form.

The above example shows the limitation of Algorithm 3. The structural result for this model show that the two step old past states and controls $(S^1_{1:t-2}, S^2_{1:t-2}, U^1_{1:t-2}, U^2_{1:t-2})$ are irrelevant to

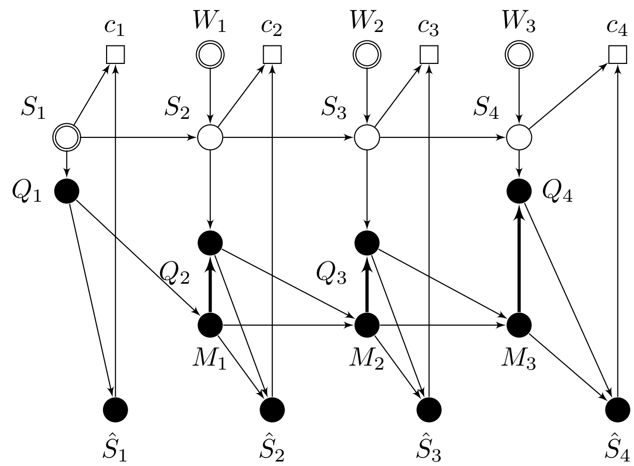


Figure 12: Simplification of the strict expansion of the team form of Figure 5. The thick lines denote the edges that were added during the strict expansion of the team form of Figure 5.

both controllers. However, Algorithm 3 cannot determine this because it is looking at irrelevant data from the point of view of only one controller. Each controller thinks that the this past data is useful to the other controller. In order to realize that this data is irrelevant to both of them, we need to consider the team form from the point of view of both controllers. An extension of Algorithm 3 that does this is presented in the next section.

5. Simplification at a group of agents

5.1. 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*

$$C_H \coloneqq \bigcap_{\alpha \in H} I_\alpha$$

and the *local information* (or private information)

$$L_{H,\alpha} \coloneqq I_\alpha \setminus C_H, \quad \alpha \in H$$

The common information is the data known to all agents in H; in fact, it is *common knowledge* (in the sense of Aumann [20]) 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.

5.2. Coordinator for a group of agents

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

1. *Variable structure*: The coordinated team form $\mathcal{F}_H$ 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

Algorithm 4: CoordinatedTeam

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 : Group of agents H
output: Coordinated team form $\mathcal{F}_H$
output: Coordinator λ_H
begin
  let λ_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 $C_H = \cap_{\alpha \in H} I_\alpha$
  $I^*_{\lambda_H} \leftarrow C_H$
  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**
    $I^*_n \leftarrow I_n$
  $\mathcal{F}_H \leftarrow \langle (N^*, A^*, B^*, M^*), \{I^*_n : n \in A^* \cup M^*\}, (K, \{D_k : k \in K\})\rangle$
  return $(\mathcal{F}_H, \lambda_H)$

unchanged. Thus,

$$A^* = A \setminus H \cup \{\lambda_H\}, \quad M^* = M \cup H, \quad B^* = B \tag{7}$$

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 $\mathcal{F}_H$ is the same as that of $\mathcal{F}$.

The procedure for obtaining a coordinated team form is shown in Algorithm 4. In addition, for any team type $\mathcal{T}$ of $\mathcal{F}$, a team type $\mathcal{T}_H$ of $\mathcal{F}_H$ 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 $n = \lambda_H$,
$$\mathbb{X}^*_{\lambda_H} = \prod_{\alpha \in H} \mathbb{Z}^\alpha_H$$
where
$$\mathbb{Z}^\alpha_H = \left(\Big(\prod_{n \in L_{H,\alpha}} \mathbb{X}_n, \prod_{n \in L_{H,\alpha}} \mathscr{F}_n\Big) \mapsto (\mathbb{X}_\alpha, \mathscr{F}_\alpha)\right), \quad \alpha \in H$$
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 dynamic*: For $m \in M$, the system dynamics are not changed, i.e.,

$$f^*_m = f_m, \quad m \in M.$$

   For $m \in H$, the system dynamics are given by

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

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

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

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 vertexes 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.,

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

We illustrate the construction of a coordinated team by a few examples.

Example 1 (Markov decision process, continued) Consider the team form of the Markov decision process shown in Figure 4 and a subset $H = \{U_3, U_4\}$ of agents. The coordinated team form $\mathcal{F}_H$ is shown in Figure 13. In this team form, $I^*_{X_{\lambda_H}} = \{S_1, S_2, S_3, U_1, U_2\}$, $I^*_{U_3} = \{X_{\lambda_H}\}$, and $I^*_{U_4} = \{X_{\lambda_H}, S_4, U_3\}$, while the information structure of the remaining nodes is the same as before. This coordinated team form corresponds to the following system: The control station at times 1 and 2 operates as before

$$\begin{aligned} U_1 &= g_1(S_1), & S_2 &= f_2(S_1, U_1, W_1), \\ U_2 &= g_2(S_{1:2}, U_1), & S_3 &= f_3(S_2, U_2, W_2). \end{aligned}$$

The coordinator at observes $(S_{1:3}, U_{1:2})$ and chooses action (Z_3, Z_4) as follows:

$$(Z_3, Z_4) = \psi(S_{1:3}, U_{1:2})$$

which are $(S_{1:3}, U_{1:2})$-sections of the control laws (g_3, g_4) in the original system. The control station at times 3 and 4 is passive and simply evaluates Z_3 and Z_4 at its local information. Thus,

$$U_3 = Z_3, \quad S_4 = f_4(S_3, U_3, W_3), \quad \text{and} \quad U_4 = Z_3(S_4, U_3).$$

Example 3 (Decentralized control with one-step delayed sharing of state, continued) Consider the team form of the decentralized control system shown in Figure 6 and a subset $H = \{U^1_4, U^2_4\}$ of agents. The coordinated team $\mathcal{F}_H$ is shown in Figure 14. In this team form $I^*_{X_{\lambda_H}} = \{S^1_{1:3}, S^2_{1:3}, U^1_{1:3}, U^2_{1:3}\}$, $I^*_{U^1_4} = \{X_{\lambda_H}\}$, and $I_{X_{U^2_4}} = \{X_{\lambda_H}\}$, while the information structure of the remaining nodes is the same as before.

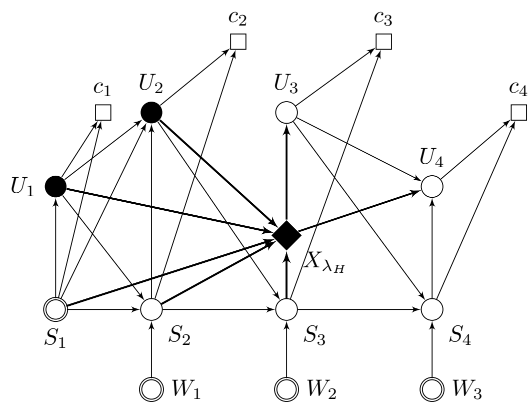


Figure 13: The coordinated team corresponding to the team form of a Markov decision process of Example 1 and a coordinator $H = \{U_3, U_4\}$. The in- and out-edges of X_{λ_H} are denoted by thick lines.

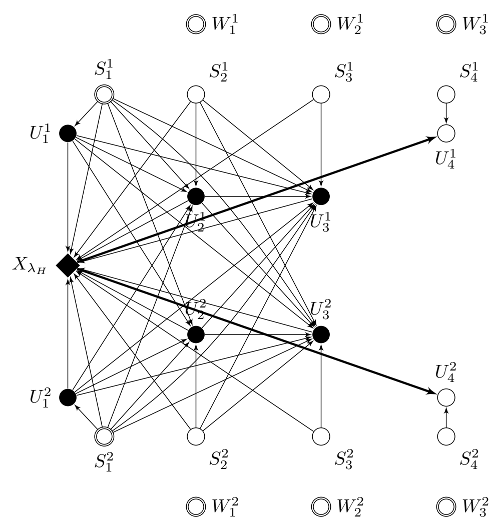


Figure 14: The coordinated team corresponding to the team form of the decentralized control system of Example 3 and a coordinator $H = \{U_4^1, U_4^2\}$. The in- and out-edges of X_{λ_H} are denoted by thick lines. For clarity, we have split the edges into two graphs. Only the information sets of the control variables are shown here. The information sets of the system variables and the cost structure is the same as Figure 6(b).

The idea of a coordinator for a group of agents was first proposed in [21] and later used in [8] to prove the structural results for delayed sharing information structures [6,7]. The coordinated team form is useful for two reasons: (i) it is equivalent to the original team form (see Proposition 3 below), and hence (ii) any structural result for the coordinated team is also applicable to the original (see Theorem 3 below). 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 $(\mathcal{F}, \mathcal{T})$ and a subset H of agents, let $(\mathcal{F}_H, \mathcal{T}_H)$ be the corresponding coordinated sequential team. Then for any policy $\mathbf{g}$ of $(\mathcal{F}, \mathcal{T})$, there exists a policy*

Algorithm 5: SimplifyGroupAt

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 : Group of agents H
output: Team form $\mathcal{F}'$
begin
  let $(\mathcal{F}_H, \lambda_H) \coloneqq$ `CoordinatedTeam(`$\mathcal{F}, H$`)`
  $(\mathcal{F}_H, J_H) \leftarrow$ `SimplifySingleAt(`$\mathcal{F}_H, \lambda_H$`)`
  forall $\alpha \in H$ **do**
    $I_\alpha \leftarrow I_\alpha \setminus J_H$
  return $\mathcal{F}$

$\mathbf{g}_H$ of $(\mathcal{F}_H, \mathcal{T}_H)$ that achieves the same cost and vice versa, i.e., for any policy $\mathbf{g}_H$ of $(\mathcal{F}_H, \mathcal{T}_H)$, there exists a policy $\mathbf{g}$ of $(\mathcal{F}, \mathcal{T})$ 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 $\mathcal{F}_H$, then J_H is also irrelevant for all $\alpha \in H$ in $\mathcal{F}$. Using this correspondence, we state the main result for a group of agents.

5.3. Main result for a group of agents

Theorem 3 *Given a team form $\mathcal{F}$ and a subset H of agents A, let $\mathcal{F}_H$ 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 $\mathcal{F}_H$, $C_H \setminus J_H \cup \{\lambda_H\}$ 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 $\mathcal{F}_H$, then, by Theorem 2, the data $\{X_n : n \in J_H\}$ is irrelevant for control at the coordinator in $\mathcal{F}_H$. 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 $\mathcal{F}$. ■

Theorem 3 suggests that to simplify a team form $\mathcal{F}$ at a group of agents H, we should look at the coordinated team $\mathcal{F}_H =$ `CoordinatedTeam(`$\mathcal{F}, H$`)` 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 $\mathcal{F}$. 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 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.

5.4. Some algorithmic remarks

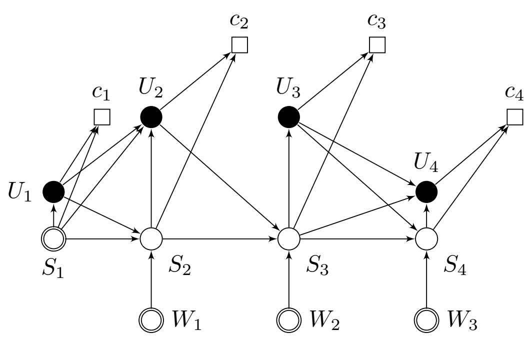
1. In Algorithm 6, we do not need to check all subsets of agents. Rather, we only need to check subsets H for which the common information $C_H \neq \emptyset$. Hence, instead of iterating for all subsets of A in line $\star$ of Algorithm 6, we only need to iterate over subsets of A for which $C_H \neq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

Algorithm 6: SimplifyTeam (Second Version)

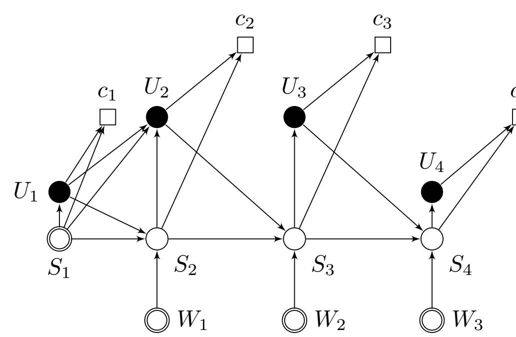
input : Team form $\mathcal{F} = \langle (N, A, B, M), \{I_n : n \in A \cup M\}, (K, \{D_k : k \in K\}) \rangle$
output: Team form $\mathcal{F}'$
begin
  $\mathcal{F} \leftarrow$ `StrictlyExpand`($\mathcal{F}$)
  repeat
    let $\mathcal{F}' := \mathcal{F}$
★   **forall** H *subset of* A **do**
      $\mathcal{F} \leftarrow$ `SimplifyGroupAt`($\mathcal{F}, H$)
  until $\mathcal{F}' = \mathcal{F}$

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.

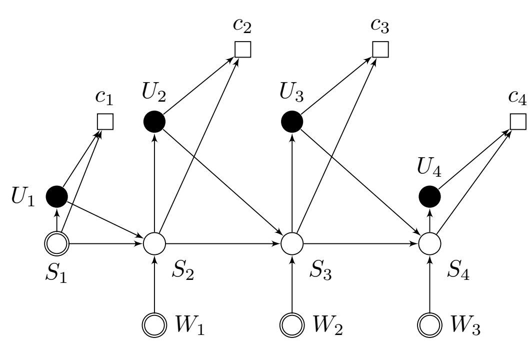
5.5. Examples



(a) $\mathcal{F}_1$



(b) $\mathcal{F}_2$



(c) $\mathcal{F}_3$

Figure 15: Different steps in the simplification of the team form of Figure 1 using Algorithm 6. $\mathcal{F}_1$ is obtained by simplifying the team form of Figure 1 at $\{U_4, U_3\}$; $\mathcal{F}_2$ by simplifying $\mathcal{F}_1$ at $\{U_4\}$ and $\mathcal{F}_3$ by simplifying $\mathcal{F}_2$ at $\{U_2\}$.

Example 1 (Markov decision process, continued) Reconsider the team form of a Markov decision process shown in Figure 4. Denote this team form by $\mathcal{F}_0$. The information structure of the team is fully expanded. Algorithm 6 on this team form proceeds as follows: We order the subsets of $A = \{U_1, U_2, U_3, U_4\}$ is reverse lexicographical order. For $H \in \{\{U_4, U_3.U_2, U_1\}, \{U_4, U_3, U_2\}, \{U_4, U_3, U_1\}, \{U_3, U_2, U1\}\}$, we have that $\texttt{SimplifyGroupAt}(\mathcal{F}_0, H) = \mathcal{F}_0$. The first simplification of $\mathcal{F}_0$ occurs for $H = \{U_4, U_3\}$. For clarity we denote $\{U_4, U_3\}$ by H_1^*. The coordinated team $\mathcal{F}_{H*_1} = \texttt{CoordinatedTeam}(\mathcal{F}_0, H_1^*)$ is shown in Figure 13. In this team form, $\{S_3, X_{\lambda_H}\}$ D-separates $J_{H_1^*} \coloneqq \{U_1, S_1, U_2, S_2\}$ from $K_{H_1^*} \coloneqq \{c_3, c_4\}$. Thus, in $\mathcal{F}_0$, we can remove the edges from $J_{H_1^*}$ to H_1^*. The simplified team form, which we denote by $\mathcal{F}_1$, is shown in Figure 15(a). Proceeding further with the team form $\mathcal{F}_1$, we have that for $H \in \{\{U_4, U_2\}, \{U_4, U_1\}, \{U_3, U_2\}, \{U_3, U_1\}, \{U_2, U_1\}\}$, $\texttt{SimplifyGroupAt}(\mathcal{F}_1, H) = \mathcal{F}_1$. Simplifying $\mathcal{F}_1$ at $\{U_4\}$ gives $\mathcal{F}_2$, which is shown in Figure 15(b). Simplifying $\mathcal{F}_2$ at $\{U_3\}$ does not change the team form; simplifying at $\{U_2\}$ gives $\mathcal{F}_3$, which is shown in Figure 15(c). Proceeding further, $\mathcal{F}_3$ does not change when we simplify it at $\{U_1\}$, or iterate once more through Algorithm 6. Hence, $\mathcal{F}_3$, shown in Figure 15, is a simplification of the team form of Figure 4 obtained by Algorithm 6.

The above example shows that applying Algorithm 3 and Algorithm 6 on the team form of Example 1 gives the same simplified team forms. This is also the case for Example 2. The advantage of Algorithm 6 is that it can also simplify team forms that are not simplified by Algorithm 3, like the team form of Example 3.

Example 3 (Decentralized control with one-step delayed sharing of state, continued) Consider the team form of the decentralized control system shown in Figure 6. Denote this team form by $\mathcal{F}_0$. The information structure of this team form is fully expanded. Consider the group $H_4 \coloneqq \{U_4^1, U_4^2\}$ of agents. The coordinated team form $\mathcal{F}_{H_4}$ is shown in Figure 14. In this team form, $\{S_3^1, S_3^2, U_3^1, U_3^2, \lambda_{H_4}\}$ D-separates $J_{H_4} \coloneqq \{S_{1:2}^1, S_{1:2}^2, U_{1:2}^1, U_{1:2}^2\}$ from $K_{H_4} \coloneqq \{c_4\}$. Thus, we can remove edges from J_{H_4} to λ_{H_4}. Removing these edges from $\mathcal{F}_{H_4}$ gives the team form shown in Figure 16(a); removing them from $\mathcal{F}_0$ gives the team form shown in Figure 16(b). We denote this latter team form by $\mathcal{F}_1$.

Consider a group $H_3 \coloneqq \{U_3^1, U_3^2\}$ of agents in $\mathcal{F}_1$. The coordinated team form is shown if Figure 16(c). In this team form $\{S_2^1, S_2^2, U_2^1, U_2^2, \lambda_{H_3}\}$ D-separates $J_{H_3} \coloneqq \{S_1^1, S_1^2, U_1^1, U_1^2\}$ from $K_{H_3} \coloneqq \{c_3, c_4\}$. Thus, we can remove the edges from J_{H_3} to λ_{H_3}. Removing these edges from $\mathcal{F}_{H_3}$ gives the team form shown in Figure 16(d); removing them from $\mathcal{F}_1$ gives the team form of Figure 17. We denote this latter team form by $\mathcal{F}_2$. Further iterations of Algorithm 6 on $\mathcal{F}_2$ do not remove any edges.

The team form $\mathcal{F}_2$, shown in Figure 17 corresponds to a system in which the information sets of U_t^i is $\{S_t^i, S_{t-1}^1, S_{t-1}^2, U_{t-1}^1, U_{t-1}^2\}$, $i = 1, 2$, $t = 2, 3, \dots$. Thus, we have recovered the structural results of one-step delayed sharing information structures using Algorithm 6.

The graphical model of Figure 6 corresponds to a team form in which the information set of U_4^i is $\{S_2^1, S_2^2, S_3^i, S_4^i, U_3^i\}$. Thus, we have recovered the structural result the structural result of [5] using Algorithm 6.

5.6. Remark

Theorems 2 and 3 suggest two techniques for identifying irrelevant information at an agent or a group of agents. These techniques, implemented algorithmically using Algorithms 3 and 6, provide a simplification of the corresponding team form. However, we have not been able to prove that the resulting team form in minimal; hence our original question (Problem 3) remains partially unanswered.

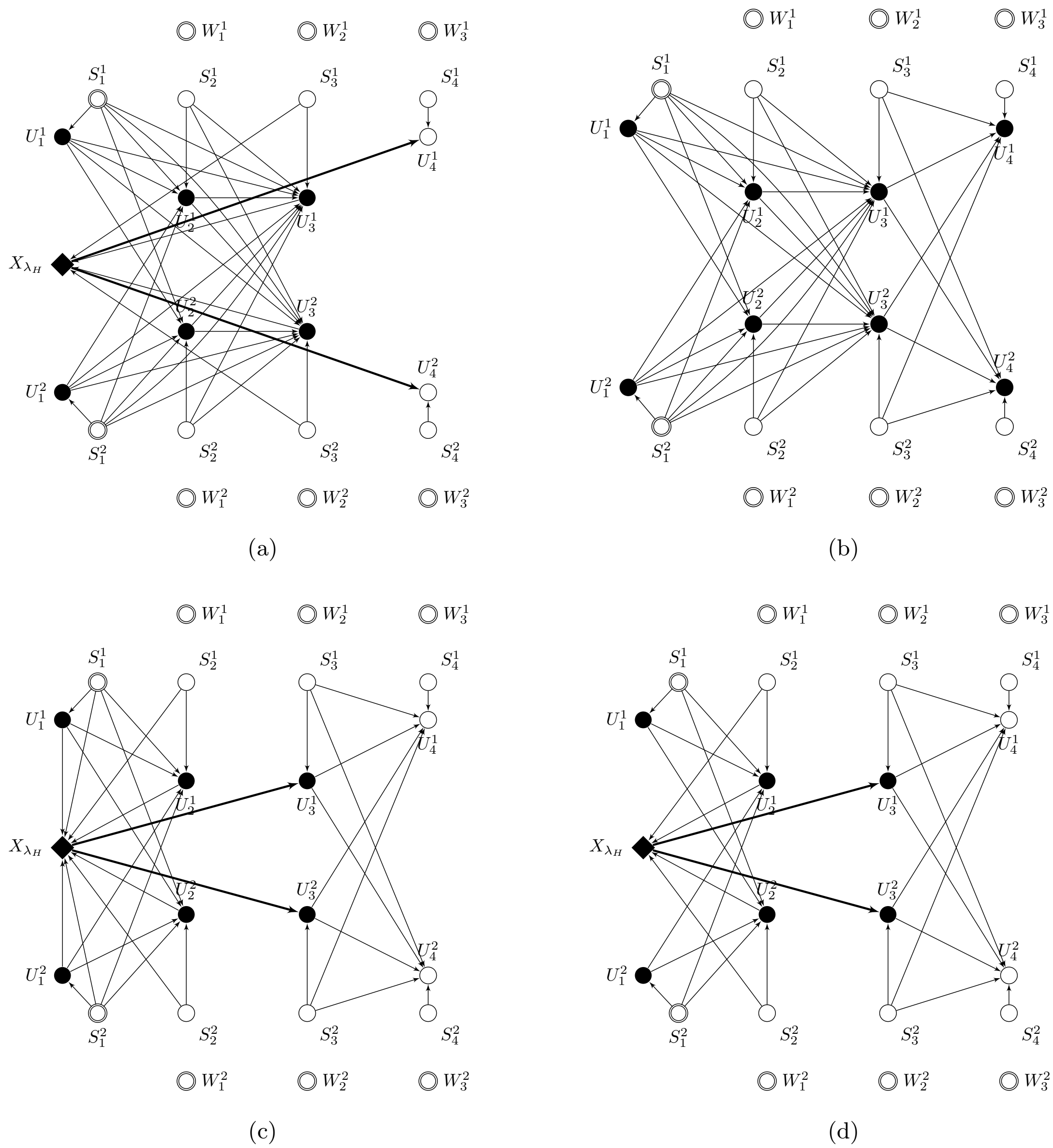


Figure 16: The intermediate team forms obtained in the simplification of the team form of Example 3 by Algorithm 6. For simplicity, we only show the information sets of the control variables. The information sets of the system dynamics and the control structure are not changed by the algorithm and remain the same as shown in Figure 6(a). The team form (a) is obtained by simplifying the coordinated team form of Figure 14, which, in turn, was obtained by the team form of Figure 6 and coordinator $H_4 \coloneqq \{U^1_4, U^2_4\}$. The team form (b) is obtained from the team form of Figure 6(a) by removing the irrelevant edges identified in (a). The team form (c) is the coordinated team form of (b) with the coordinator $H_3 \coloneqq \{U^1_3, U^2_3\}$. The team form (d) is a simplification of (c). Removing the corresponding edges from (b) gives the team form shown in Figure 17.

6. Conclusion

In this paper, we present a framework for the algorithmic derivation of structural results of sequential teams.A key concept for this framework is the notion of a team form. This notion captures the

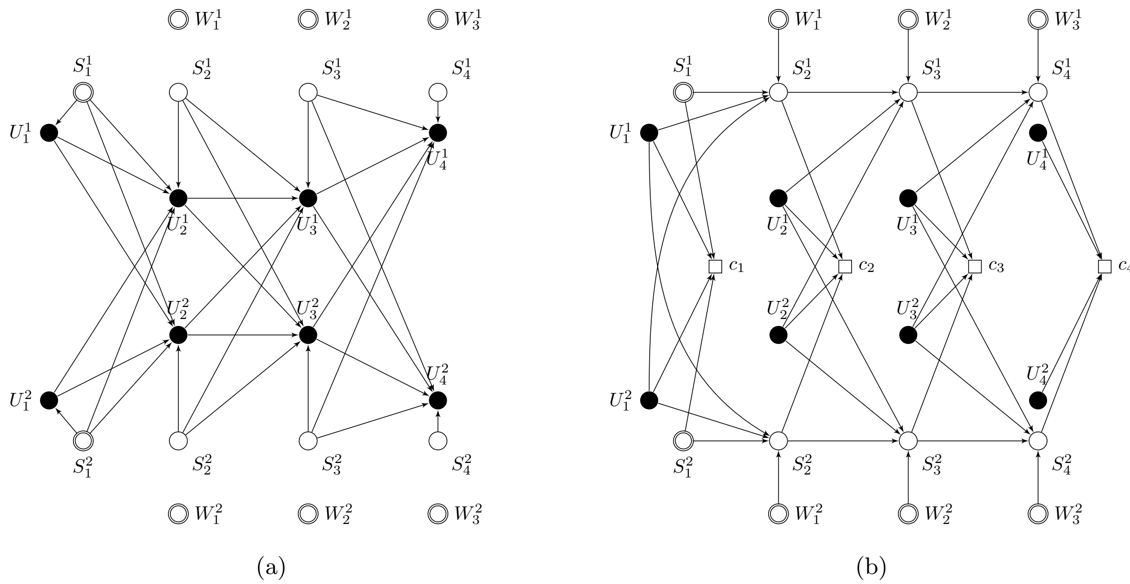


Figure 17: The final simplified team form of Example 3 obtained by Algorithm 6

properties of a sequential team that do not depend on the specifics of the state spaces, probability measure, and the cost function. We represent a team form as a DAG (directed acyclic graph) and show that structural results of a sequential team is equivalent to removing edges from this DAG. We present algorithms that identify which edges of the graph may be removed; thereby providing an algorithmic framework for deriving structural results. At the same time, we believe that the proposed framework is a useful pedagogical tool for understanding and visualizing the structural results of sequential teams.

Acknowledgments

The authors are grateful to Achilleas Anastasopoulos, Ashutosh Nayyar, Maxim Raginsky, and Demosthenis Teneketzis for discussions and comments.

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

$$\mathcal{F}_\ell = \{(N, A, B, M), \{I_n^\ell, n \in A \cup M\}, (K, \{D_k, k \in K\})\}, \quad \ell = 0, \dots, L$$

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 $\ell = 0, \dots, L$, $n \in A \cup M$,

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

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 $(\mathcal{F}, \mathcal{T})$ and $(\mathcal{F}', \mathcal{T})$ have the value.

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 $(\mathcal{F}_\ell, \mathcal{T})$. Next we will show that the reverse implication is also true.

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_ℓ} 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

$$g_\alpha(I_\alpha) = \begin{cases} g'_\alpha(I_\alpha), & \alpha \neq \alpha_\circ \\ g'_{\alpha_\circ}(I_{\alpha_\circ}, \psi(I_{\alpha_\circ}; \mathbf{g}')), & \alpha = \alpha_\circ \end{cases} \tag{8}$$

By construction, for any realization of the primitive random variables $\{X_\beta : \beta \in B\}$, using the policy $\mathbf{g}$ given by (8) 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 $(\mathcal{F}_{\ell+1}, \mathcal{T})$ 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 $\mathcal{F}_L = \mathcal{F}'$.

Appendix B. Bayes ball algorithm for identifying graph irrelevant nodes

The Bayes ball algorithm [16] checks for D-separation in a graphical model and identifies graphically irrelevant nodes. For the sake of completeness, we present the algorithm, refined to the graphical model corresponding to the team form, here.

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 $\alpha_\circ \in A^*$,

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

Algorithm 7: GraphIrrelevant (the Bayes Ball Algorithm)

```
input  : Team form F = ⟨(N, A, B, M), {I_n : n ∈ A ∪ M}, (K, {D_k : k ∈ K})⟩
input  : Observed data O
input  : Target Set T
output: Irrelevant data O'
begin
    let (V, S, R_*, R^*, P_*, P^*) = (∅, T, T, ∅, N, N)
    while S ≠ ∅ do
        Pick s ∈ S
        (V, S) ← (V ∪ {s}, S \ {s})
        if s ∈ R_* then
            if s ∈ O then
                // Do nothing.  The ball is blocked
            else
                // Pass the ball through
                if s ∈ P^* then
                    P^* ← P^* \ {s}
                    S ← S ∪ N_G^-(s)
                if s ∈ B ∩ P_* then
                    P_* ← P_* \ {s}
                    S ← S ∪ N_G^+(s)
        if s ∈ R^* then
            if s ∈ O then
                // Bounce back the ball
                if s ∈ P^* then
                    P^* ← P^* \ {s}
                    S ← S ∪ N_G^-(s)
            else
                // Pass the ball through
                if s ∈ P_* then
                    P_* ← P_* \ {s}
                    S ← S ∪ N_G^+(s)
    return O \ V
```

where

$$\psi_H^\alpha : \Big(\prod_{n \in C_H} \mathbb{X}_n, \prod_{n \in C_H} \mathscr{F}_n \Big) \mapsto \left(\Big(\prod_{n \in L_{H,\alpha}} \mathbb{X}_n, \prod_{n \in L_{H,\alpha}} \mathscr{F}_n \Big) \mapsto (\mathbb{X}_\alpha, \mathscr{F}_\alpha) \right), \quad \alpha \in H$$

which is given by

$$\psi_H^\alpha(\cdot) = g_\alpha(\cdot, \{X_n : n \in C_H\}), \quad \alpha \in H$$

where $g_\alpha(\cdot, \{X_n : n \in C_H\})$ 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 $\alpha_\circ \in A$,

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

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

[1] J. Marschak and R. Radner, *Economic Theory of Teams*. New Haven: Yale University Press, 1972.

[2] Y.-C. Ho, "Team decision theory and information structures," *Proc. IEEE*, vol. 68, no. 6, pp. 644–654, 1980.

[3] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation Identification and Adaptive Control*. Prentice Hall, 1986.

[4] H. S. Witsenhausen, "On the structure of real-time source coders," *Bell System Technical Journal*, vol. 58, no. 6, pp. 1437–1451, July-August 1979.

[5] M. Aicardi, F. Davoli, and R. Minciardi, "Decentralized optimal control of Markov chains with a common past information set," *IEEE Trans. Autom. Control*, vol. 32, no. 11, pp. 1028–1031, 1987.

[6] H. S. Witsenhausen, "Separation of estimation and control for discrete time systems," *Proc. IEEE*, vol. 59, no. 11, pp. 1557–1566, Nov. 1971.

[7] P. Varaiya and J. Walrand, "On delayed sharing patterns," *IEEE Trans. Autom. Control*, vol. 23, no. 3, pp. 443–445, 1978.

[8] A. Nayyar, A. Mahajan, and D. Teneketzis, "Optimal control strategies in delayed sharing information structures," *accepted for publication inIEEE Trans. Autom. Control*, 2011.

[9] D. Blackwell, "Memoryless strategies in finite-stage dynamic programming," *Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 863–865, 1964.

[10] P. Whittle, *Optimization Over Time*, ser. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons, 1983, vol. 2.

[11] H. S. Witsenhausen, "The intrinsic model for discrete stochastic control: Some open problems," in *Control Theory, Numerical Methods and Computer System Modelling*, ser. Lecture Notes in Economics and Mathematical Systems, A. Bensoussan and J. L. Lions, Eds. Springer Verlag, 1975, vol. 107, pp. 322–335.

[12] ——, "Equivalent stochastic control problems," *Mathematics of Control Signals and Systems*, vol. 1, pp. 3–11, 1988.

[13] ——, “On information structures, feedback and causality,” *SIAM Journal of Control*, vol. 9, no. 2, pp. 149–160, May 1971.

[14] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes.* Oxford University Press, USA, 2001.

[15] D. Geiger, T. Verma, and J. Pearl, “Identifying independence in Bayesian networks,” *Networks*, vol. 20, no. 5, pp. 507–534, 1990.

[16] R. D. Shachter, “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*, 1998, pp. 480–487.

[17] Y.-C. Ho and K.-C. Chu, “Team decision theory and information structures in optimal control problems–Part I,” *IEEE Trans. Autom. Control*, vol. 17, no. 1, pp. 15–22, 1972.

[18] T. Yoshikawa, “Decomposition of dynamic team decision problems,” *IEEE Trans. Autom. Control*, vol. 23, no. 4, pp. 627–632, Aug. 1978.

[19] S. Adlakha, S. Lall, and A. Goldsmith, “A Bayesian network approach to control of networked Markov decision processes,” in *Proceedings of the 46th Allerton conference on communication*, 2008.

[20] R. J. Aumann, “Agreeing to disagree,” *Annals of Statistics*, no. 4, pp. 1236–39, 1976.

[21] A. Mahajan, A. Nayyar, and D. Teneketzis, “Identifying tractable decentralized control problems on the basis of information structures,” in *proceedings of the 46th Allerton conference on communication, control and computation*, Sep. 2008, pp. 1440–1449.