

# Finding the weakest link(s): Coalition games for decentralized discrete-event control

Laurie Ricker, Hervé Marchand

► **To cite this version:**

Laurie Ricker, Hervé Marchand. Finding the weakest link(s): Coalition games for decentralized discrete-event control. 55th IEEE Conference on Decision and Control , Dec 2016, Las-Vegas, United States. 2016. <hal-01373709>

**HAL Id: hal-01373709**

**<https://hal.inria.fr/hal-01373709>**

Submitted on 6 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Finding the weakest link(s): Coalition games for decentralized discrete-event control

S.L. Ricker<sup>1</sup> and H. Marchand<sup>2</sup>

**Abstract**—In decentralized discrete-event system (DES) architectures, agents fuse their local decisions to arrive at the global decision. The contribution of each agent to the final decision is never assessed; however, it may be the case that only a subset of agents, i.e., a (static) coalition, perpetually contribute towards the correct final decisions. In casting the decentralized DES control (with and without communication) problem as a cooperative game, it is possible to quantify the average contribution that each agent makes towards synthesizing the overall correct control strategy. Specifically, we explore allocations that assess contributions of non-communicating and communicating controllers for this class of problems. This allows a quantification of the contribution that each agent makes to the coalition with respect to decisions made solely based on its partial observations and decisions made based on messages sent to another agent(s) to facilitate a correct control decision.

## I. INTRODUCTION

In the classic formulation of decentralized control of DES [1], [2], the contribution each controller makes towards the global control decision is considered “equal” in the sense that as long as at least one controller, based on its partial observation of the system, can take the correct control decision, the overall correct control strategy can be synthesized. With the introduction of synchronous communication between controllers (e.g., [3]), again, no one controller is valued over another, as long as the communication protocol allows, as before, at least one controller to take the correct control decision, leading to the correct global decision.

Yet controllers are not necessarily equal in such a decision-making architecture. When assigning a quantitative measure to the process of reaching a control decision, the partial observation of each controller coupled with the information that it might be required to send and/or receive to allow the collection of controllers to reach the correct control decisions may render some controllers more “useful” than others. To that end, we are interested in the problem of coalition formation for decentralized controllers.

A solution concept from coalition games, the Shapley value [4], provides one measure for assessing the contributions of a set of agents that cooperate to reach a specified goal. If reaching a correct control decision is assigned a value, then it is possible to allocate the total value of

a complete control solution among the decentralized controllers, where those that contributed to correct disablement decisions are rewarded. Subsequently, we are interested in casting the decentralized control problem (in the absence of communication) as a coalition game, and using the Shapley value to determine which subsets of decentralized controllers are best combined to reach the control objective.

When a communication network plays a role in a coalition game, the Myerson value [5] is the first allocation rule of interest, where, like the Shapley value, each participant in the game receives a payoff based on their contribution to achieving the overall goal. Unfortunately, the calculation of this value is ultimately insensitive to the nuances of the network formation. When decentralized controllers communicate, the overall cost to solve the control problem is predicated on the specific communication protocol amongst the players. Thus, an appropriate allocation rule must take the configuration of the protocol into consideration.

A variety of other allocation rules have been proposed in the domain of network games as alternatives to the Myerson value. The most notable is the Link-Based Flexible network allocation rule [6], which rewards those links that prove to be vital to the success of the game and allows for the network to change, whenever the possibility for such flexibility arises. We adapt a recently-proposed allocation rule for dynamic networks [7] to the DES control and (synchronous) communication problem. This new rule allows us to take into account both the evolving network configuration and the ability to synthesize a control and communication policy among a group of controllers, some of whom are not always active participants in the decision making.

Game theory has provided useful models for the analysis of discrete-event control problems: Nash equilibrium of decentralized controllers [8] and synchronous communication protocols [9]; fictitious play to examine optimal decentralized controllers for intrusion detection [10]; Pareto optimality of decentralized communicating controllers [11]. Furthermore, coalitional game theory is starting to play a significant role in evaluating network configurations in continuous control theory [12], [13].

We review some concepts from decentralized control in Section II. Section III introduces the Shapley value as applied to the decentralized control problem without communication. We also attune a new quantitative metric from dynamic network games to assess the value of dynamically-forming coalitions of communicating decentralized controllers, before concluding in Section IV.

\*This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)

<sup>1</sup>S.L. Ricker is with Dept. of Mathematics & Computer Science, Mount Allison University, Sackville, NB, Canada [lricker@mta.ca](mailto:lricker@mta.ca)

<sup>2</sup>H. Marchand is at INRIA Rennes-Bretagne Atlantique, Campus de Beaulieu, Rennes, France [herve.marchand@inria.fr](mailto:herve.marchand@inria.fr)

## II. BACKGROUND

We assume the framework for supervisory control of discrete-event systems as introduced in [14]. Hence, we model a discrete-event system as a finite-state automaton

$$M_L = (Q, \Sigma, T_L, q_0, F),$$

where  $Q$  is a finite set of states,  $T_L \subseteq Q \times \Sigma \times Q$  is a transition relation,  $q_0 \in Q$  is the initial state, and  $F \subseteq Q$  is a set of final or marked states. (We will also use a version of  $M_L$  that contains self-loops of the empty string  $\varepsilon$  at each state in  $Q$ :  $M_L^\varepsilon = (Q, \Sigma \cup \{\varepsilon\}, T_L^\varepsilon, F)$ .) The transition relation is easily extended to  $\Sigma^*$  and we say  $L := \{s \in \Sigma^* \mid \exists q \in Q \text{ s.t. } q_0 \xrightarrow{s} q \in T_L\}$ . For  $L \subseteq \Sigma^*$ , we have  $\bar{L} := \{v \in \Sigma^* \mid \exists w \in \Sigma^*, u \in L \text{ such that } u = vw\}$ . Then  $L$  is *prefix-closed* if  $L = \bar{L}$ . We assume prefix-closed languages for the remainder of this paper. The regular language  $L$  generated by  $M_L$  describes the behavior of the uncontrolled system. In addition, the alphabet  $\Sigma$  is the disjoint union of two types of events: (i) observable and unobservable events  $\Sigma = \Sigma_o \uplus \Sigma_{uo}$ ; and (ii) controllable and uncontrollable events  $\Sigma = \Sigma_c \uplus \Sigma_{uc}$ .

### A. Decentralized control

The decentralized control problem arises when controllers, denoted here by  $I = \{1, \dots, n\}$ , where  $n > 1$ , cooperate by fusing their local control decisions about behaviors (e.g., disable or enable a specific event) to issue global control decisions so that a specification language  $K \subseteq L$  is recognized (where we have the corresponding automaton  $M_K \subseteq M_L$ ). The marked language of  $L$  is  $L_m := \{s \in L \mid \exists q \in F \text{ s.t. } q_0 \xrightarrow{s} q \in T_L\}$ . The language  $K \subseteq L$  is *m-closed* if  $K = \bar{K} \cap L_m$ . Henceforth, we assume that  $K$  is *m-closed*.

In the original formulation of the decentralized control problem [1], there is no communication among controllers. In the event that a control solution cannot be found in one of the various decentralized architectures [1], [15], there are conditions under which decentralized communication protocols can be synthesized, allowing synchronously communicating controllers to reach a control decision (e.g., [3], [16], [17]).

A decentralized control problem is characterized by each controller's set of local events that it observes  $\Sigma_{o,i} \subseteq \Sigma_o$  and a set of local events it controls  $\Sigma_{c,i} \subseteq \Sigma_c$ , for  $i \in I$ . The *natural projection*  $\pi_i : \Sigma^* \rightarrow \Sigma_{o,i}^*$  defines the observations of an agent  $i \in I$  by removing all occurrences of events in  $\Sigma \setminus \Sigma_{o,i}$  from a sequence in  $\Sigma^*$ . The inverse projection  $\pi_i^{-1} : \Sigma_{o,i}^* \rightarrow 2^{\Sigma^*}$  captures all the sequences  $s$  that produce the same natural projection for agent  $i$ . Formally,  $\pi_i^{-1}(t) = \{s \in \Sigma^* \mid \pi_i(s) = t\}$ , for  $i \in I$ , and can easily be extended to languages. We will use the notation  $\llbracket s \rrbracket_i$  whenever we refer to  $\pi_i^{-1}\pi_i(s)$ . For further notational convenience, we let  $I_o(\sigma) = \{i \in I \mid \sigma \in \Sigma_{o,i}\}$ , and  $I_c(\sigma) = \{i \in I \mid \sigma \in \Sigma_{c,i}\}$ .

**Definition 1:** ([14]) *A language  $K \subseteq L$  is **controllable** wrt  $L$  and  $\Sigma_{uc}$  iff  $\bar{K} \Sigma_{uc} \cap L \subseteq \bar{K}$ .*

Thus, an uncontrollable event cannot take the system out of the specification.

As the object of the control problem is to prevent sequences in  $L \setminus \bar{K}$  from occurring, we would like at least one

decentralized controller, based only on its partial observation, to be able to definitively identify when a sequence in  $L \setminus \bar{K}$  is about to occur [1].

**Definition 2:** *A language  $K \subseteq L = \bar{L}$  is **unconditionally co-observable** with respect to  $L$ ,  $\Sigma_{o,i}$  and  $\Sigma_{c,i}$  ( $i \in I$ ) iff  $(\forall t \in \bar{K})(\forall \sigma \in \Sigma_c) s\sigma \in L \setminus \bar{K} \Rightarrow (\exists i \in I_c(\sigma)) \llbracket s \rrbracket_i \sigma \cap \bar{K} = \emptyset$ . Note that when  $I = \{1\}$  (i.e., a centralized supervisor), this property is called **observability**.*

### B. Synchronized composition

We use a special product called *synchronized composition*, denoted by  $\times_S$ , which is defined as follows. Assume that we have  $k > 1$  finite-state automata  $M_1, \dots, M_n$ , where  $M_j = (Q_j, \Sigma_j, T_j, q_{0,j}, F_j)$ , for  $j = 1, 2, \dots, k$ . Then  $M = M_1 \times_S M_2 \times_S \dots \times_S M_k = (Q_S, \Sigma_S, T_S, \langle q_{0,1}, q_{0,2}, \dots, q_{0,k} \rangle, F_S)$ , where  $Q_S \subseteq Q_1 \times Q_2 \times \dots \times Q_k$ ;  $\Sigma_S \subseteq \Sigma_1 \times \Sigma_2 \times \dots \times \Sigma_k$ ;  $T_S \subseteq Q_S \times \Sigma_S \times Q_S$ ; and  $F_S \subseteq F_1 \times F_2 \times \dots \times F_k$ . The state set  $Q_S$  is a set of state vectors of the form  $q_S = (q_1, \dots, q_k)$  and we will occasionally refer to the  $j^{\text{th}}$  component of  $q_S$  as  $q_S(j)$ , where  $j \in \{1, \dots, k\}$ . We can think of these automata as running concurrently, and, thus, there is some synchronization of events in each of the alphabets. The alphabet  $\Sigma_S$  is constructed using vector labels [18].

We build a structure using  $\times_S$  where the components are  $n + 1$  copies of  $M_L^\varepsilon$  and the alphabet  $\mathbb{A}$ , composed of both observable  $\mathbb{A}_o$  and unobservable subsets of labels  $\mathbb{A}_{uo}$ , is the set of vector labels defined above:  $\mathcal{U} = M_L^\varepsilon \times_S \dots \times_S M_L^\varepsilon = (X, \mathbb{A}, T_{\mathcal{U}}, x_0)$ .  $\mathcal{U}$  is a multi-purpose structure: we use it for verifying co-observability and to construct our controllers. To accomplish the former, we identify transitions of  $\mathcal{U}$  that correspond to violations of co-observability, namely a state where a controllable event  $\sigma$  must be disabled in the uncontrolled system, but all controllers that control  $\sigma$ , believe that the system has generated (up to that point) a sequence after which  $\sigma$  must be enabled.

**Definition 3:** *The set of **unconditional illegal configurations** for  $\mathcal{U}$  is a set of transitions defined as follows.*

$$\begin{aligned} \mathcal{B}_V &:= \{x \xrightarrow{\ell} x' \in T_{\mathcal{U}} \mid x(0) \xrightarrow{\ell(0)=\sigma} x'(0) \in T_L \setminus T_K \\ &\quad \text{and } (\forall i \in I_c[\ell(0)]) x(i) \xrightarrow{\ell(i)} x'(i) \in T_K\}. \end{aligned}$$

When  $\mathcal{B}_V = \emptyset$ , we know that  $K$  is unconditionally co-observable [17]. To synthesize the decentralized controllers, under the condition that  $\mathcal{B}_V = \emptyset$ , we calculate the *crushed automaton* of  $\mathcal{U}$  w.r.t. each controller, which requires the following equivalence relation on the state set of  $\mathcal{U}$ .

**Definition 4:** ([19]) *Let  $\Omega = \mathbb{A}$ . The relation  $\sim_\Omega$  is the least equivalence on the set  $Q$  such that the following two axioms are satisfied*

$$\begin{aligned} \text{A1: } & (x, \ell, x') \in T_{\mathcal{U}} \text{ and } \ell \notin \Omega \Rightarrow x \sim_\Omega x'; \\ \text{A2: } & (x_1, \ell', x'_1) \in T_{\mathcal{U}} \text{ and} \\ & (x_2, \ell', x'_2) \in T_{\mathcal{U}} \text{ and } x_1 \sim_\Omega x_2 \Rightarrow x'_1 \sim_\Omega x'_2. \end{aligned}$$

The associated **crushed automaton** is  $\kappa_\Omega[\mathcal{U}] = (X/\sim_\Omega, [x_0]_\Omega, \Omega, T_{\kappa_\Omega})$ , where  $[x_0]_\Omega$  is the equivalence class of the initial state  $x_0$ ; and  $([x]_\Omega, \ell, [x']_\Omega) \in T_{\kappa_\Omega}$  if there

are states  $x_1$  and  $x'_1$  such that  $x_1 \sim_\Omega x$ ,  $x'_1 \sim_\Omega x'$  and  $(x_1, \ell, x'_1) \in T_{\mathcal{U}}$ .

We construct a crushed automaton of  $\mathcal{U}$  for each controller  $i \in I$  using  $\Omega_i = \{\ell \in \mathbb{A} \mid \ell(0) = \ell(i) \in \Sigma_{o,i}\}$ , and  $\Omega_0 = \mathbb{A}_o$ . For acyclic  $M_L$ ,  $\kappa_{\Omega_0}[\mathcal{U}]$  is isomorphic to  $M_L$ , whereas when  $M_L$  is cyclic, our solutions require an unfolding of the plant up to  $\kappa_{\Omega_0}[\mathcal{U}]$ . Note that a crushed automaton is based on state equivalences and its construction has a computational complexity of  $O(|X| + |T_{\mathcal{U}}|)$ .

### C. State-based decentralized decision-making

In our control architecture, we use  $\mathcal{U}$  for decentralized decision-making. We classify the states of  $\mathcal{U}$  as follows:  $\mathbf{B}(\sigma) := \{x \in X \mid \exists x' \in X \text{ s.t. } (x(0), \sigma, x'(0)) \in T_L \setminus T_K \wedge \sigma \in \Sigma_c\}$ , where  $\mathbf{B}_\sigma = \cup_{\sigma \in \Sigma_c} \mathbf{B}(\sigma)$ ;  $\mathbf{G}(\sigma) := \{x \in X \mid \exists x' \in X \text{ s.t. } (x(0), \sigma, x'(0)) \in T_K\}$ .

Each controller  $i \in I$  makes its own local control decisions about the system behavior based on its partial observations via  $\sim_{\Omega_i}$ . Let the finite set of local decisions  $\mathbf{LD} = \{0, 1\}$ . Let the local decision function be  $h_i : X \rightarrow \mathbf{LD}^\Sigma$

$$(\forall i \in I)(\forall x \in X)(\forall \sigma \in \Sigma)$$

$$h_i([x]_{\Omega_i})(\sigma) = \begin{cases} 1, & \text{if } \sigma \in \Sigma_{c,i} \wedge [x]_{\Omega_i} \cap \mathbf{G}(\sigma) = \emptyset; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

To correctly solve the decentralized DES problem of interest, a global decision  $\mathbf{GD} = \{0, 1\}$  is made by combining the local decisions of the controllers in such a way that the correct decision is taken overall<sup>1</sup>. Let the global decision function be  $H : \mathbf{LD}^{\Sigma^n} \rightarrow \mathbf{GD}^\Sigma$ .

We want to find local decision functions  $h_1, \dots, h_n$  such that the global decision function  $H$  satisfies the following:

$$\begin{aligned} (\forall \sigma \in \Sigma_c)(\forall x \in \mathbf{B}(\sigma)) \\ H([x]_{\Omega_0})(\sigma) &= \bigvee_{i \in I_c(\sigma)} h_i([x]_{\Omega_i})(\sigma) = 1; \\ (\forall \sigma \in \Sigma_c)(\forall x \in \mathbf{G}(\sigma)) \\ H([x]_{\Omega_0})(\sigma) &= \bigvee_{i \in I_c(\sigma)} h_i([x]_{\Omega_i})(\sigma) = 0; \\ (\forall \sigma \in \Sigma_{uc})(\forall x \in \mathbf{G}(\sigma)) \\ H([x]_{\Omega_0})(\sigma) &= \bigvee_{i \in I_c(\sigma)} h_i([x]_{\Omega_i})(\sigma) = 0. \end{aligned}$$

We can find such local decision functions (e.g., synthesize decentralized controllers) for  $L$  iff  $K$  is controllable,  $m$ -closed, and co-observable.<sup>2</sup>

<sup>1</sup>We focus on unconditional co-observability [1]; however our results are easily applied to other decentralized architectures. Here we require disablement decisions to produce a positive (non-zero) value. Hence, we use disjunction as the fusion rule in lieu of conjunction.

<sup>2</sup>When communication is involved, replace  $H$  by  $H^?$ ,  $h_i$  by  $h_i^?$  and  $\Omega_i$  by  $\Omega_i^?$  (for  $i \in I_0$ ). Note that  $\Omega_0 = \Omega_0^?$  since the 0th ‘‘agent’’ has observation set  $\Sigma_o$  and no access to any communications sent or received by the decentralized agents.

### D. Decentralized control with synchronous communication

When decentralized controllers have the ability to communicate among themselves (e.g., point-to-point, broadcast), they do so via a communication protocol. We adopt the strategy introduced in [17] where subsets of observable transitions are identified as *possible communication transitions* corresponding to messages to be sent from sender/controller  $i$  to receiver/controller  $j$ , denoted by  $T_{ij}^{!?} \subseteq T_{o,i}$  for  $i, j \in I$ .

We rely on an architectural property of  $\mathcal{U}$  that provides a straightforward means of identifying potential communication transitions with which we build a communication protocol, denoted by  $\mathcal{T}^! \subseteq \cup_{i,j \in I} T_{ij}^{!?}$ :

**Definition 5:** (Adapted from [22].) *The diamond/step property holds at  $x_1 \in X$  if there exist  $i, j \in I$  and labels  $\ell_1 \in \mathbb{A}_o$ ,  $\ell_2 \in \mathbb{A}_{uo}$  such that  $\ell_1(0) = \ell_1(i) = \gamma \in \Sigma_{o,i}$ ,  $\ell_2(j) = \gamma \neq \varepsilon$  and  $\ell_1 \vee \ell_2 \in \mathbb{A}$  that satisfy the following axioms: (a)  $x_1 \xrightarrow{\ell_1} x_2$ ,  $x_1 \xrightarrow{\ell_2} x_3 \in T_{\mathcal{U}} \wedge \ell_1 | \ell_2 \Rightarrow (\exists x_4 \in X) A x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T_{\mathcal{U}}$ ; (b)  $x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T_{\mathcal{U}} \wedge \ell_1 | \ell_2 \Rightarrow (\exists x_2 \in X) x_1 \xrightarrow{\ell_1} x_2, x_2 \xrightarrow{\ell_2} x_4 \in T_{\mathcal{U}}$ ; (c)  $x_1 \xrightarrow{\ell_1} x_2, x_2 \xrightarrow{\ell_2} x_4 \in T_{\mathcal{U}} \wedge \ell_1 | \ell_2 \Rightarrow x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4 \in T_{\mathcal{U}}$ . Subsequently, we say that at  $x_1$  we have a *possible communication transition*  $x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4$  w.r.t. sender  $i$  and receiver  $j$ .*

Choosing a communication transition to add to a communication protocol means that we must remove or *prune*  $x_1 \xrightarrow{\ell_1} x_2$  and  $x_1 \xrightarrow{\ell_2} x_3$  from  $\mathcal{U}$ . Furthermore, we must also remove these transitions at all states that are equivalent to  $x_1$  for the sender:

$$\begin{aligned} (\forall x \in [x_1]_{\Omega_i}) \exists x'' \in X \text{ s.t. } x \xrightarrow{\ell_1} x'' \in T_{\mathcal{U}} \\ \Rightarrow T_{\mathcal{U}} = T_{\mathcal{U}} \setminus \{x \xrightarrow{\ell_1} x''\}; \end{aligned}$$

and the receiver

$$\begin{aligned} (\forall x \in [x_1]_{\Omega_j}) \exists x'' \in X \text{ s.t. } x \xrightarrow{\ell_2} x'' \in T_{\mathcal{U}} \\ \Rightarrow T_{\mathcal{U}} = T_{\mathcal{U}} \setminus \{x \xrightarrow{\ell_2} x''\}. \end{aligned}$$

We also update the communication protocol for communicating controller  $i$ :  $\mathcal{T}_{ij}^! = \mathcal{T}_{ij}^! \cup \{x_1 \xrightarrow{\ell_1 \vee \ell_2} x_4\}$ .

The choice of a communication transition is not random: we choose a communication transition so that, after pruning, an illegal configuration in  $\mathcal{B}_\vee$  becomes unreachable. There are a finite number of illegal configurations to eliminate and when all of these are removed from  $\mathcal{U}$ , then we conclude that unconditional co-observability has been restored via communication [17].

**Definition 6:** *Two states  $x, x' \in X$  are indistinguishable to supervisor  $i$ , denoted  $x \sim_i x'$ , where  $\sim_i$  is the least equivalence relation such that*

- (1)  $x \xrightarrow{\langle \ell(0), \dots, \ell(i) = \varepsilon, \dots, \ell(n) \rangle} x' \Rightarrow x \sim_i x'$ ;
- (2)  $x \xrightarrow{\langle \varepsilon, \dots, \varepsilon, \ell(i) \neq \varepsilon, \varepsilon, \dots, \varepsilon \rangle} x' \Rightarrow x \sim_i x'$ ;
- (3)  $x \sim_i x' \wedge (x, \ell, x''), (x', \ell, x''') \in T_{\mathcal{U}} \Rightarrow x'' \sim_i x'''$ .

In addition to allowing the communicating decentralized supervisors to make the correct control decisions, a com-

munication protocol must operate in an observationally-equivalent manner.

**Definition 7:** A communication protocol  $\mathcal{T}^1 = \cup_{ij \in I, i \neq j} \mathcal{T}_{ij}^1$  is **feasible** iff  $(\forall i, j \in I)(\forall x, x' \in X) (x, \ell, x'') \in \mathcal{T}_{ij}^1 \wedge (x', \ell, x''') \in T^U \wedge x \sim_i x' \Rightarrow (x', \ell, x''') \in \mathcal{T}_{ij}^1$ .

Thus, a protocol is feasible if whenever supervisor  $i$  sends a message to supervisor  $j$  at state  $x$ , the same communication occurs at all states that supervisor  $i$  considers observationally equivalent to  $x$ . Henceforth, we assume that all communication protocols we discuss are feasible.

### III. DECENTRALIZED DES AS COALITIONAL GAMES

A coalitional game with a transferable utility is a pair  $(I, v)$ , where  $I$  is a *set of players* and  $v$  is a *characteristic function* that provides a quantitative measure, which can be used to compare the effectiveness of different coalitions [20]. In particular, the interest lies in which coalition of players  $\mathcal{C} \subseteq I$  will form and how  $v$  will be distributed to the members of  $\mathcal{C}$ . The *grand coalition* is the name given to the coalition of all agents in  $I$ . Subsequently, we define the characteristic function  $v : \mathcal{C} \rightarrow \mathbb{R}$  that assigns a real number representing the quality of the coalition's efforts. Notably,  $v(\emptyset) = 0$ . A solution to a coalitional game is a method for dividing  $v(I)$  among the players.

#### A. Games without networks

An important solution concept for cooperative games, called the *Shapley value*, defines a weighted average of each player's marginal contribution across different coalitions.

**Definition 8:** ([4]) *The Shapley value* for player  $i \in I$ , denoted by  $\phi_i$ , is a *measure of the relative importance of every player in a coalition game*:

$$\phi_i(I, v) = \frac{1}{|I|!} \sum_{\mathcal{C} \subseteq I \setminus \{i\}} |\mathcal{C}|!(|I| - |\mathcal{C}| - 1)! [v(\mathcal{C} \cup \{i\}) - v(\mathcal{C})].$$

While the computation of the Shapley value requires exhaustive enumeration of coalitions, efficient algorithms have recently been proposed [21].

We will use the Shapley value to assess the average contribution of each controller to all possible coalitions. Because the characteristic function is superadditive, the assumption is that players/controllers will form the grand coalition, as it provides the optimal allocation of value for  $I$ .

Although there are many different characteristic functions that could be designed for decentralized DES control problems, for simplicity, we recognize the combination of the coalition's information that corresponds to the overall correctness of the control solution:

$$v(\mathcal{C}) = \sum_{x \in X} \sum_{\sigma \in \Sigma_{\mathcal{C}}} H_{\mathcal{C}}([x]_{\Omega_0})(\sigma),$$

where  $H_{\mathcal{C}}([x]_{\Omega_0})(\sigma) = \bigvee_{i \in \mathcal{C}} h_i([x]_{\Omega_i})(\sigma)$ .

While the Shapley value—as interpreted for decentralized DES—indicates the contribution of each controller towards the overall control solution, this information could be incorporated into a variety of controller-related metrics.

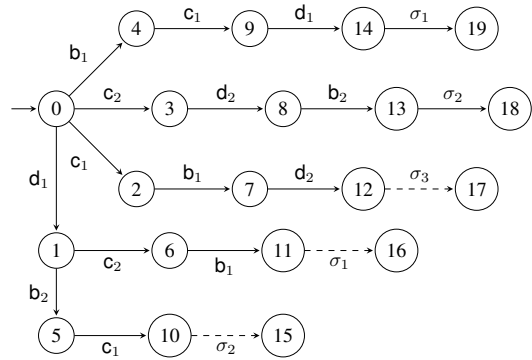


Fig. 1. Joint  $M_L$  (all transitions) and  $M_K$  (collection of solid transitions)

For example, to support a certain level of fault tolerance, the characteristic function can be adjusted to guide the selection of coalitions that provide necessary redundancy for particularly important control decisions. In the event of controller breakdown or temporary unavailability, are there other controllers that can correctly take the control decisions that the now absent controller took? Conversely, we can use the Shapley value to deduce that a controller is not necessary. For costly controllers, it may be important to determine the cheapest set of controllers that can solve the control problem. Similarly, identifying the smallest set of controllers (i.e., minimally optimal cardinality) that take all the correct control decisions may be a performance priority.

**Example 1:** We will use  $M_L$  and  $M_K$  in Fig. 1 to construct  $\mathcal{U}$  (not shown) and subsequently examine different decentralized non-communicating coalitions. Let  $I = \{1, 2, 3\}$  where  $\Sigma_{o,1} = \{\mathbf{b}_1, \mathbf{b}_2, \sigma_1, \sigma_2, \sigma_3\}$ ,  $\Sigma_{o,2} = \{\mathbf{c}_1, \mathbf{c}_2, \sigma_1, \sigma_2, \sigma_3\}$  and  $\Sigma_{o,3} = \{\mathbf{d}_1, \mathbf{d}_2, \sigma_1, \sigma_2, \sigma_3\}$ . Furthermore, let  $\Sigma_{c,1} = \Sigma_{c,2} = \Sigma_{c,3} = \{\sigma_1, \sigma_2, \sigma_3\}$ .

It is the case that  $K$  is co-observable (Definition 2). This can also be verified using  $\mathcal{U}$ :  $(\forall \sigma \in \Sigma_c)(\forall x \in X) x \in \mathbf{B}(\sigma) \Rightarrow (\exists i \in I_c(\sigma))[x]_{\Omega_i} \cap \mathbf{G}(\sigma) = \emptyset$ .

We are interested in the following states where disablement decisions must be taken:  $(11, 11, 11, 11)$ ,  $(11, 11, 11, 14)$ ,  $(11, 14, 11, 11)$ ,  $(11, 14, 11, 14) \in \mathbf{B}(\sigma_1)$ ,  $(10, 10, 10, 10)$ ,  $(10, 13, 10, 10) \in \mathbf{B}(\sigma_2)$  and  $(12, 12, 12, 12) \in \mathbf{B}(\sigma_3)$ . Note that  $\{(11, 11, 11, 11), (11, 11, 11, 14), (11, 14, 11, 11), (11, 14, 11, 14)\} \subset [(11, 11, 11, 11)]_{\Omega_i}$ , for all  $i \in I$ . Similarly,  $\{(10, 10, 10, 10), (10, 13, 10, 10)\} \subset [(10, 10, 10, 10)]_{\Omega_i}$  and  $\{(12, 12, 12, 12)\} \subseteq [(12, 12, 12, 12)]_{\Omega_i}$ , for all  $i \in I$ .

Using Eq. (1) to calculate local control decisions, there is at least one controller that can take the correct disablement decision for  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$ :  $h_2([(11, 11, 11, 11)]_{\Omega_2})(\sigma_1) = 1$ ,  $h_2([(10, 10, 10, 10)]_{\Omega_2})(\sigma_2) = 1$ ,  $h_3([(10, 10, 10, 10)]_{\Omega_3})(\sigma_2) = 1$ , and, for all  $i \in I$ ,  $h_i([(12, 12, 12, 12)]_{\Omega_i})(\sigma_3) = 1$ .

In Table I we calculate the characteristic function for all possible coalitions, where there are three disablement decisions to take. Although not shown in Table I, the order of the coalition does not matter, i.e.,  $v(\{2, 3\}) = v(\{3, 2\})$ .

TABLE I  
 $v(\mathcal{C})$  FOR  $\mathcal{C} \subseteq \{1, 2, 3\}$

$\mathcal{C}$	{1}	{2}	{3}	{1, 2}	{1, 3}	{2, 3}	{1, 2, 3}
$v(\mathcal{C})$	1	3	2	3	3	3	3

To calculate the Shapley value of controller 1, we must consider coalitions that do not include it:  $\emptyset$ ,  $\{2\}$ ,  $\{3\}$  and  $\{2, 3\}$ .

$$\begin{aligned} \phi_1(I, v) &= \frac{1}{6} [2(v(\{1\}) - v(\emptyset)) + (v(\{1, 2\}) - v(\{2\})) \\ &\quad + (v(\{1, 3\}) - v(\{3\})) \\ &\quad + 2(v(\{1, 2, 3\}) - v(\{2, 3\}))] = \frac{1}{2}. \end{aligned}$$

The Shapley values of the remaining controllers:  $\phi_2(I, v) = \frac{9}{6}$ ;  $\phi_3(I, v) = 1$ . Note that the sum of the Shapley values is equal to the value of the grand coalition. For this particular example, controller 2, on average, contributes the most value to the control solution. If minimizing the cardinality of the size of a coalition is an important metric, while still ensuring that all the correct control decisions are taken, then the optimal coalition is  $\mathcal{C} = \{2\}$ .  $\square$

### B. Games with networks

When players require communication to coordinate their activities, the organization of the channels that define how the players communicate is described by a *network*. A coalition  $\mathcal{C}$  in this context is characterized by a sequence of configurations of the network, where not all players may be connected depending on how the network evolves. A network  $\lambda \subseteq \{ij \mid i, j \in I \wedge i \neq j\}$  is simply a set of links  $ij$  that indicates which players are connected. The empty network  $\lambda^\emptyset$  is one in which there are no connections between players, whereas the complete network  $\lambda^I$  is simply the binomial coefficient  $\binom{n}{2}$ , i.e., the set of all possible subsets of  $I$  of size 2. The symmetric difference of two networks  $\lambda$  and  $\lambda'$ :  $\lambda \Delta \lambda' = (\lambda \cup \lambda') \setminus (\lambda \cap \lambda')$ . Unlike the previous situation, the coalition here is not static from the start of the execution of the system to its end: the membership changes based on the communication requirements needed to take the correct control decisions.

One of the criticisms of the Myerson value is that it is insensitive to the specific communication network formed by the elements of  $\lambda$ . It is possible to find examples where two different communication networks among the same set of players produces the same final allocation. In the context of a decentralized DES framework, we have two extensions to the network model that we must address. It may not be that communication is required to ensure all the correct control decisions are taken. Specifically, a single controller may have sufficient knowledge to take the correct control decision in the absence of communication. So while the definition of  $\lambda$  precludes a link of  $ii$ , we will allow  $\lambda$  to be coupled with, when necessary, single controllers in  $I$  that are not already part of a link in  $\lambda$ . Secondly, we must identify the links available for communication between controllers, and also the message(s) that must be sent from the sender to the

receiver. We extend the definition of network to include the messages being sent across the links  $\lambda$ . That is, the *enhanced* network that we will work with is  $\lambda^! \subseteq (\lambda \cup I) \times (T^! \cup \emptyset)$ , where  $\emptyset$  represents the absence of messages sent along a given link in  $\lambda$ .

To take into account the evolution of a network, say step by step over a finite horizon  $W$ , we construct a sequence of enhanced networks, called a *scenario*:  $\Lambda^! = \lambda_0^!, \lambda_1^!, \dots, \lambda_{W-1}^!$ . For the decentralized DES framework, we let

$$W = \max_{x \in \mathcal{B}(\sigma)} \mathbf{d}(x_0, x),$$

where  $\mathbf{d}(x_1, x_2)$  is the “distance” between states in terms of the number of observable labels along an acyclic path from state  $x_1$  to  $x_2$  in  $\mathcal{U}$ . We denote the change in the network from step  $k$  to  $k+1$  by  $\lambda_k^! \rightarrow \lambda_{k+1}^!$ . An *elementary* change is made when the  $\lambda_{k+1}^!$  differs by only a single link from  $\lambda_k^!$ , i.e., ignore the messages but focus on how the set of senders and receivers changes from one step to the next. When a change is not elementary, then it is necessary to compute the sequence of elementary changes that takes the links in  $\lambda_k^!$  to those in  $\lambda_{k+1}^!$ .

We construct  $\Lambda$  that describes, over all the violations of unconditional co-observability, the dynamics of the extended network formation. It is necessary to describe a condition under which we can ensure that despite the changing membership of the active decision makers, all the correct control decisions can still be taken.

*Definition 9:*  $\mathcal{U}$  is  $\Lambda$ -**observable** iff  $\forall x \in \mathcal{B}(\sigma) \exists \lambda^! \in \Lambda$  such that  $[x]_{\Omega \lambda^!} \cap \mathcal{G}(\sigma) = \emptyset$ .

We propose two different strategies for forming scenarios that produce a  $\Lambda$ -observable version of  $\mathcal{U}$ : myopic and farsighted. For the former strategy, we want to choose the first possible communication opportunity to resolve a violation of unconditional co-observability. A farsighted strategy exercises the opposite approach: postpone communication as long as possible. Algorithm 1 employs a myopic approach to ensure  $\Lambda$ -observability; however, to adapt the algorithm to be farsighted, simply change “minimal” to “maximal”, with the understanding that we are always assuming maximal acyclic distances.

The myopic algorithm proceeds as follows: for each disablement state in  $\mathcal{U}$ , we want to find a the scenario that will allow at least one controller to take the correct disablement decision before the system leaves  $\overline{K}$ . If the disablement state is part of an illegal configuration, then we want to find the “earliest” possible communication transition that will result in the illegal configuration being unreachable. In our context, “earliest” refers to a state where the diamond/step property holds, the illegal configuration can be resolved, and the number of observable events that occur from the initial state to a state where the diamond/step property holds is minimal. The sender is a controller that observes the event encoded in the communication transition and the receiver is a controller capable of taking the control decision for the illegal configuration in question. At this point we have identified the extended network at this step (i.e., the number

of observable events) along with the message. We must then identify all feasible communications and incorporate them into the current scenario. If the disablement state is not part of an illegal configuration, then we simply identify any of the controllers capable of taking the control decision and add them to the extended network as a non-communicating participant.

Once we have identified the scenarios for  $\mathcal{U}$ , we can calculate the value that will be distributed among the decentralized controllers.

*Definition 10:* (adapted from [7]) *The link-based scenario allocation rule for dynamic network processes, over a finite horizon  $W$ , is given by:*

$$\phi^\Lambda(v) = \sum_{k=0}^{W-1} (\phi_i^{\lambda_k^! \rightarrow \lambda_{k+1}^!}(v))_{i \in I}, \quad (2)$$

where, for an elementary change of scenarios,

$$\phi_i^{\lambda_k^! \rightarrow \lambda_{k+1}^!}(v) = \begin{cases} 1/2(v(\lambda_{k+1}^!) - v(\lambda_k^!)), & \text{if } (ij, \cdot) \in \lambda_{k+1}^! \Delta \lambda_k^!, \\ v(\lambda_{k+1}^!), & \text{if } (i, \emptyset) \in \lambda_{k+1}^! \Delta \lambda_k^!, \\ 0, & \text{otherwise.} \end{cases}$$

If the change between scenarios is not an elementary one, the computation of  $\phi_i^{\lambda_k^! \rightarrow \lambda_{k+1}^!}(v)$  is significantly more complex and requires the computation of a weighted average over the transformation of the non-elementary change to a sequence of elementary changes [7]. The example presented below features scenarios with only elementary changes, and thus we restrict the discussion to the more straightforward version of the allocation rule.

To apply the link-based value to our setting, we assume that the uncontrolled system does not satisfy co-observability, so we must construct feasible communication transitions  $\mathcal{T}^!$  and ensure that the system is coalition-observable over  $\Lambda$ . Thus, the value of our coalition for any given network configuration is determined by the number of correct control decisions that can be taken when  $\lambda^!$  is the communication protocol:

$$v(\lambda^!) = \sum_{x \in X} \sum_{\sigma \in \Sigma_c} H^{\lambda^!}([x]_{\Omega_0})(\sigma),$$

where  $H^{\lambda^!}([x]_{\Omega_0})(\sigma) = \bigvee_{i \in \lambda^!} h_i^{\lambda^!}([x]_{\Omega_i^{\lambda^!}})(\sigma)$ , and  $\Omega_i^{\lambda^!}$  is the local alphabet induced by  $\lambda^!$  for controller  $i$ .

Intuitively, the addition of a communication produces a value of at least 1. This characteristic function does not impose a penalty for the evolution of the network and any costs that might be associated with changing communication links. Thus, this version of  $v$  assumes that all communication between controllers is possible. In this case we can use our result to minimize the number of communications between controllers, differing from the quantitative strategy introduced in [17]. In this case we could simply fix a static extended network  $\lambda^!$  (where all scenarios  $\Lambda$  are simple finite sequences of the same  $\lambda^!$ ) and directly use  $\phi^\Lambda(v)$  (or perhaps a minor adaptation) to determine an optimal communication strategy. This remains the subject of future investigation.

*Example 2:* We will use  $M_L$  and  $M_K$  in Fig. 2 to build  $\mathcal{U}$ , and to examine the link-based network allocation value of a decentralized DES with communicating controllers.

Let  $\Sigma_{o,1} = \{\mathbf{a}, \sigma_1, \sigma_2, \sigma_3\}$ ,  $\Sigma_{o,2} = \{\mathbf{b}, \sigma_1, \sigma_2, \sigma_3\}$  and  $\Sigma_{o,3} = \{\mathbf{c}, \mathbf{d}, \sigma_1, \sigma_2, \sigma_3\}$ . Furthermore, let  $\Sigma_{c,3} = \{\sigma_3\}$  and  $\Sigma_{c,1} = \Sigma_{c,2} = \{\sigma_1, \sigma_2, \sigma_3\}$ . This system does not satisfy co-observability (Definition 2).

The full  $\mathcal{U}$  structure for this example has 20857 states, 2835 potential communications to include in our set of communications  $T^!$ , 18 states where disablement decisions must be taken, including 2 violations of unconditional co-observability. The complete paths to the violations are shown in Fig. 3. Of interest is the presence of potential communication transitions (colored blue), which represents the circumstance that communication chosen here is ill-advised. Pruning at either  $(0, 0, 1, 1)$  to select a communication of  $\mathbf{c}$  from controller 3 to controller 2, or  $(3, 0, 4, 4)$  to select a communication of  $\mathbf{b}$  from controller 2 to controller 3, does not make the illegal configuration at  $(15, 10, 10, 10)$  unreachable. Thus, it is not the case that all communications are useful.

We run Algorithm 1 on  $\mathcal{U}$  using the myopic strategy where  $W = 4$ , starting with violations in  $\mathbf{B}(\sigma_1)$ , corresponding to the occurrence of  $\mathbf{cba}$  after which  $\sigma_1$  must be disabled. For the myopic strategy, we choose a communication of the first occurrence of  $\mathbf{c}$  where the sender is controller 3 and we actually do not care whether the receiver is controller 1 or controller 2. For the farsighted strategy, we choose the first occurrence of  $\mathbf{b}$  that controller 2 will communicate to controller 1. Note that we do not choose for controller 1 to communicate the occurrence of  $\mathbf{a}$  to controller 2, as this information arrives to controller 2 too late for it to avoid taking anything but the incorrect control decision (e.g., enabling  $\sigma_1$ ).

Thus, we have the following myopic scenario, where the notation has been adjusted to reflect the relevant event that requires disablement during the scenario execution:  $\Lambda_{\sigma_1}^M = \lambda_0^! = (\lambda^0, \emptyset)$ ,  $\lambda_1^! = (\{31\}, \mathbf{c})$ ,  $\lambda_2^! = (\lambda^0, \emptyset)$ ,  $\lambda_3^! = (\lambda^0, \emptyset)$ . In contrast, we have the following farsighted scenario:  $\Lambda_{\sigma_1}^F = (\lambda^0, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ,  $(\{21\}, \mathbf{b})$ ,  $(\lambda^0, \emptyset)$ .

Similarly we have the remaining scenarios:

- $\Lambda_{\sigma_2}^M = (\lambda^0, \emptyset)$ ,  $(\{21\}, \mathbf{b})$ ,  $(\lambda^0, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ;
- $\Lambda_{\sigma_2}^F = (\lambda^0, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ,  $(\{12\}, \mathbf{a})$ ,  $(\lambda^0, \emptyset)$ ;
- $\Lambda_{\sigma_3}^M = (\lambda^0, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ,  $(\{3\}, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ;
- $\Lambda_{\sigma_3}^F = (\lambda^0, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ,  $(\{3\}, \emptyset)$ ,  $(\lambda^0, \emptyset)$ ;

We use Eqn (2) to calculate the distribution of value among the controllers for each scenario. In both cases all three controllers are involved in ensuring that the correct control solution is produced; however, the distribution varies depending on whether we favour a myopic or farsighted strategy. Note that there is a valid solution not reached by either strategy that eliminates the participation of controller 3: the decision for disabling  $\sigma_3$  is made by either having controller 2 communicate its observation of  $\mathbf{b}$  to controller 1 or controller 1 communicating its observation of  $\mathbf{a}$  to controller 2. In this case, all correct control decisions can

---

**Algorithm 1** Forming dynamic networked coalitions and  $\Lambda$ -observable  $\mathcal{U}$ .

---

```

1: procedure FORMMYOPICDYNAMICCOALITIONS( $\mathcal{U}$ ,  $\{\mathbf{B}(\sigma)\}_{\sigma \in \Sigma_c}$ ,  $\mathcal{B}_V$ )
2:   for each  $\sigma \in \Sigma_c$  do
3:     for each state  $x \in \mathbf{B}(\sigma)$  do
4:       Initialize  $\Lambda_\sigma^! = \{\lambda_k^! \leftarrow (\lambda^\emptyset, \emptyset)\}_{k=0, W-1}$ 
5:       if  $x$  is involved in a violation in  $\mathcal{B}_V$  then
6:         while there are still violations in  $\mathcal{B}_V$  involving  $x$  do
7:           Choose  $(x'', \ell', x''') \in T^!?$  that resolves violation  $(x, \ell, x') \in \mathcal{B}_V$  such that distance  $\mathbf{d}(x_0, x'')$  is
           minimal
8:           Let sender  $i \in I$  be chosen such that  $\ell'(i) \in \Sigma_{\sigma, i}$ 
9:           Let receiver  $j \in I \setminus i$  be chosen such that  $j \in I_c(\sigma)$ 
10:           $\mathcal{T}_{ij}^! \leftarrow \mathcal{T}_{ij}^! \cup \{(x'', \ell', x''')\} \cup \text{FEASIBLE}(x'', \ell', x''')$ 
11:          Add  $\lambda_{ij}^! \leftarrow (ij, \ell')$  to  $\Lambda_\sigma^!$ 
12:          Add feasible communications to  $\Lambda_\sigma^!$ 
13:         else
14:           Choose  $i \in I_c(\sigma)$  that can take the correct control decision for  $\sigma$  at  $x$ 
15:           Let  $\mathbf{d}(x_0, x)$  be the minimal distance from  $x_0$  to  $x$ 
16:           Add  $\lambda_i^! \leftarrow (i, \emptyset)$  to  $\Lambda_\sigma^!$ 

```

---

TABLE II  
RESULTS OF LINK-BASED ALLOCATION RULE FOR EXAMPLE 2.

	Controller 1	Controller 2	Controller 3
$\Lambda_{\sigma_1}^M$	0.5	0	0.5
$\Lambda_{\sigma_2}^M$	0.5	0.5	0
$\Lambda_{\sigma_3}^M$	0	0	1
Totals	1	0.5	1.5
$\Lambda_{\sigma_1}^F$	0.5	0.5	0
$\Lambda_{\sigma_2}^F$	0.5	0.5	0
$\Lambda_{\sigma_3}^F$	0	0	1
Totals	1	1	1

be taken, and each receives a value of 1.5, while controller 3 receives 0, keeping in mind that only members of the extended network can take control decisions. If the cost of establishing communication between controllers 1 and 2 is less than that of bringing controller 3 into the extended network, then this would be the optimal choice. At the moment, Algorithm 1 favours non-communicating controllers who can take the correct control decision over choosing to establish a communication to facilitate the control decision. This is because no illegal configuration exists in this case, so the motivation for communication is not clear.

#### IV. CONCLUSION

We have presented an application of game theory to our ongoing study of quantitative evaluations of decentralized DES control. It may not be prudent to continue assuming that all controllers should be treated equally in terms of their value to the computation of the overall control solution. Coalition games—with and without networked communication—provide quantitative valuations that allow a comparison of the relative contributions of each controller. Such information is beneficial, in particular for applications where the cardinality and/or overall cost of the controllers is a concern, or when fault tolerance is required.

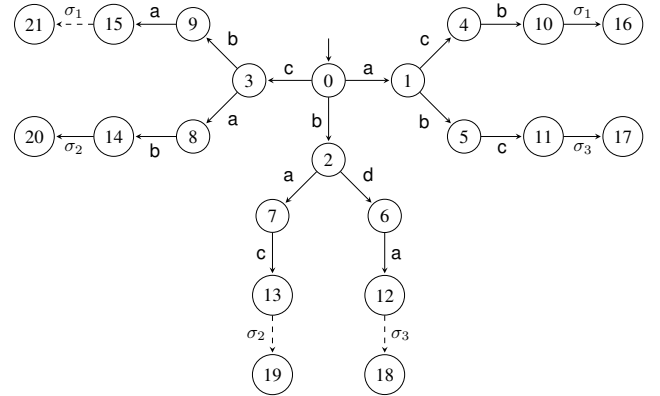


Fig. 2. Joint  $M_L$  (all transitions) and  $M_K$  (collection of solid transitions)

#### REFERENCES

- [1] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1692–1708, 1992.
- [2] T.-S. Yoo and S. Lafontaine, "A general architecture for decentralized supervisory control of discrete-event systems," *Discrete Event Dyn. S.*, vol. 12, no. 3, pp. 335–377, 2002.
- [3] K. C. Wong and J. H. van Schuppen, "Decentralized supervisory control of discrete-event systems with communication," in *Proc. Int. Workshop on Discrete Event Systems*, 1996, pp. 284–289.
- [4] L. S. Shapley, "A value for  $n$ -person games," in *Contributions to the Theory of Games*, R. Kuhn and A. Tucker, Eds. Princeton University Press, 1953, pp. 307–317.
- [5] R. Myerson, "Graphs and cooperations in games," *Mathematics of Operations Research*, vol. 2, pp. 225–229, 1977.
- [6] M. O. Jackson, "Allocation rules for network games," *Games and Economic Behavior*, vol. 51, no. 1, pp. 128–154, 2005.
- [7] J.-F. Caulier, M. Grabisch, and A. Rusinowska, "An allocation rule for dynamic random network formation processes," *Economic Theory*, vol. 60, no. 2, pp. 282–313, 2015.
- [8] A. Overkamp and J. H. van Schuppen, "Maximal solutions in decentralized supervisory control," *SIAM J. Contr. Optim.*, vol. 39, no. 2, pp. 492–511, 2000.
- [9] W. H. Sadid, S. L. Ricker, and S. Hashtrudi-Zad, "Nash equilibrium for communication protocols in decentralized discrete-event systems,"



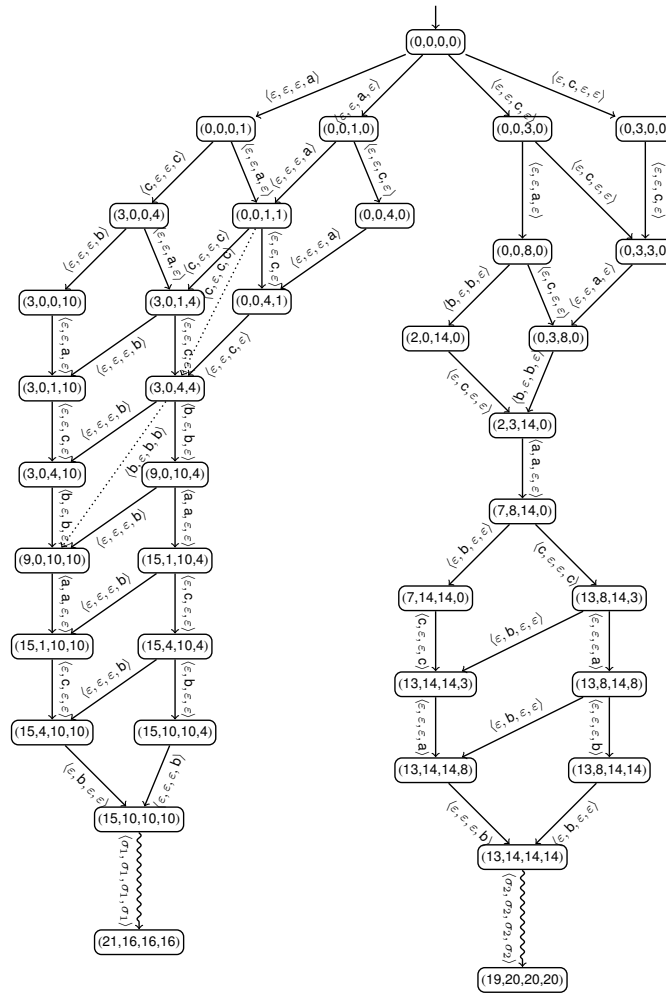


Fig. 3. A portion of  $\mathcal{U}$  showing all paths to the illegal configurations (wavy transitions) for Example 2. Dotted transitions represent potential communication transitions.

in *Proceedings of the American Control Conference*, 2010, pp. 3384–3389.

[10] D. P. L. Thorsley, “Application of stochastic techniques to partially observed discrete event systems,” Ph.D. dissertation, University of Michigan, 2006.

[11] W. H. Sadid, S. L. Ricker, and S. Hashtrudi-Zad, “Multiobjective optimization in control with communication for decentralized discrete-event systems,” in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 372–377.

[12] J. Maestre, D. M. de la Pena, A. J. Losada, E. A. Duran, and E. Camacho, “A coalitional control scheme with applications to cooperative game theory,” *Optimal Control Applications and Methods*, vol. 35, no. 5, pp. 592–608, 2014.

[13] J. Marden and J. Shamma, “Game theory and distributed control,” in *Handbook of Game Theory*, P. Young and S. Zamir, Eds. Elsevier Science, 2013, vol. 4.

[14] P. J. Ramadge and W. M. Wonham, “Supervisory control of a class of discrete event processes,” *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[15] T.-S. Yoo and S. Lafontaine, “Decentralized supervisory control with conditional decisions: supervisor existence,” *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 1886–1904, 2004.

[16] G. Barrett and S. Lafontaine, “Decentralized supervisory control with communicating controllers,” *IEEE Trans. Automat. Control*, vol. 45, no. 9, pp. 1620–1638, 2000.

[17] S. L. Ricker, “Asymptotic minimal communication for decentralized discrete-event control,” in *Proceedings of the 9th International Workshop on Discrete Event Systems*, Goteburg, Sweden, 2008, pp. 486–491.

[18] A. Arnold, *Finite transition systems*. Prentice-Hall, 1994.

[19] R. Morin, “Decompositions of asynchronous systems,” in *CONCUR: LNCS 1446*, 1998, pp. 549–564.

[20] D. Ross, “Game theory,” in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., 2016, forthcoming URL = <http://plato.stanford.edu/archives/spr2016/entries/game-theory/>.

[21] O. Skibski, T. P. Michalak, T. Rahwan, and M. Wooldridge, “Algorithms for Shapley and Myerson values in graph-restricted games,” in *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, A. Lomuscio, P. Scerri, A. Bazzan, and M. Huhns, Eds., 2014, pp. 197–204.

[22] V. Diekert and G. Rozenberg, Eds., *The Book of Traces*. World Scientific, 1995.