2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Multiobjective Optimization in Control with Communication for Decentralized Discrete-Event Systems

W.H. Sadid, S.L. Ricker and S. Hashtrudi-Zad

Abstract— **The trade-off between the cost and accuracy of a decentralized discrete-event control solution with synchronously communicating controllers is explored as a multiobjective optimization problem. We examine a class of problems where communication protocols are necessary to realize the exact control solution. In certain circumstances, it may be advantageous, from a cost perspective, to reduce communication, but incur a penalty for synthesizing an approximate control solution. A widely-used evolutionary algorithm (NSGA-II) is adapted to examine the set of Pareto-optimal solutions that arise for this family of decentralized discrete-event systems.**

I. INTRODUCTION

Quantitative optimal control has been examined from the perspective of centralized discrete-event control [9], [12], [21]. Costs are assigned to control decisions, and the goal is to synthesize a controller with an overall minimal cost with respect to the control strategy. An alternate technique for measuring the cost of centralized control was introduced in [16]. Although not developed with control theory applications in mind, a new class of quantitative languages (based on weighted automata) has also been proposed [3]. Optimal decentralized control in the absence of communication, using Nash equilibrium as the optimization criterion, was studied in [13]. In [23] the notion of fictitious play is employed to find quantitatively optimal decentralized control strategies for an intruder/detection problem. An algorithm for calculating Nash equilibrium of multi-agent systems was adapted for the quantitative analysis of communication protocols in decentralized discrete-event control [20].

We are interested in a class of quantitative decentralized discrete-event control problems where we want to optimize more than one function or *objective*: giving rise to a *multiobjective optimization* problem [22]. When incorporating communication into the decentralized control problem, there may be a cost advantage to synthesizing only part of the specification, instead of realizing the entire specification with a costly communication protocol. To that end we want to investigate the trade-off between the cost of an exact control solution achieved with communication and an approximate solution, where penalties are assessed for achieving a sublanguage of a desired controllable and observable specification, with a possibly cheaper communication policy. We examine our multiobjective optimization problem using evolutionary algorithms [1].

W.H. Sadid and S. Hashtrudi-Zad are with the Dept. of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada `m_sadid, shz@ece.concordia.ca`.

L. Ricker is with the Dept. of Mathematics & Computer Science, Mount Allison University, Sackville, NB, Canada `lricker@mta.ca`. This work has been partially supported by NSERC.

Evolutionary algorithms, inspired by biological processes, are ideal for optimization problems when (i) exhaustive search is computationally prohibitive, and (ii) there are multiple objectives to optimize. An initial population of possible solutions are considered, and a measure of their *fitness* determines whether a member of the population will be involved in the formulation of the next generation of the population. Just as in natural adaptation, over a period of many generations, a population of solutions evolve that are "closer" to an optimal solution than their predecessors. We use a modified version of the *Non-dominated Sorting Genetic Algorithm* (NSGA-II) [5], which has already proven useful for a diverse range of control problems (e.g., [7], [24]).

The paper is organized as follows. The next section contains terminology and notation of discrete-event systems, decentralized admissible control laws and communication protocols. Decentralized control and communication costs are defined in Section III to present the multiobjective optimization problem in decentralized discrete-event systems. An example is given in this section to find the trade-off between the cost and precision of a control solution using an evolutionary algorithm.

II. BACKGROUND

Supervisory control of discrete-event systems proposed in [15] uses formal language theory to model the behavior of an uncontrolled system as well as the desired behavior (*specification*) for the controlled system. Specifically, the system behavior is described by a regular language L, which can be represented by a finite automaton, M_L:

$$M_L = (Q, \Sigma, \delta, q_0, Q_{\mathsf{m}}),$$

where Q is a finite set of states; Σ is a finite set of symbols called the *alphabet*; δ is the transition function defined as $\delta : Q\times\Sigma \to Q$ and we write $\delta(q,\sigma)!$ when $\exists q' \in Q$ such that $\delta(q,\sigma) = q'$; q_0 is the initial state; and $Q_{\mathsf{m}} \subseteq Q$ is a set of marked states. The specification is denoted by the language K, where $K \subseteq L \subseteq \Sigma^*$, and an automaton marking K is M_K. Finite automata are also used to model the controllers that issue commands to ensure that the system adheres to a given specification. We are interested in systems where n controllers (let $I = \{1,\ldots,n\}$) independently issue control commands to ensure that the specification is met.

The *prefix closure* of a language K is defined as $\overline{K} := \{s \in \Sigma^* \mid (\exists t \in \Sigma^*) \text{ such that } st \in K\}$. When K is prefix-closed $K = \overline{K}$. The *marked* language of M_L, denoted by

L_{m}, is defined as $L_{\mathsf{m}} := \{s \in L \mid \delta(q_0, s) = q' \wedge q' \in Q_{\mathsf{m}}\}$. K is said to be L_{m}-closed if $K = \overline{K} \cap L_{\mathsf{m}}$.

The synchronous product of two automata $M_i = (Q_i, \Sigma_i, \delta_i, q_{0i}, Q_{\mathsf{m}i})$, for $i = \{1, 2\}$, is denoted by $M_1 || M_2$ and is defined as the reachable subgenerator of the automaton $M_1 \times M_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_{1||2}, (q_{01}, q_{02}), Q_{\mathsf{m}1} \times Q_{\mathsf{m}2})$, where

$$\delta_{1||2}((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)), & \text{if } \delta_1(q_1, \sigma)! \wedge \\ & \delta_2(q_2, \sigma)! \\ & \wedge \sigma \in \Sigma_1 \cap \Sigma_2; \\ (\delta_1(q_1, \sigma), q_2), & \text{if } \delta_1(q_1, \sigma)! \wedge \\ & \sigma \in \Sigma_1 \setminus \Sigma_2; \\ (q_1, \delta_2(q_2, \sigma)), & \text{if } \delta_2(q_2, \sigma)! \wedge \\ & \sigma \in \Sigma_2 \setminus \Sigma_1; \\ \textit{undefined}, & \text{otherwise.} \end{cases}$$

In the context of the supervisory control problem, Σ is partitioned into two sets for each controller: *controllable* events that can be prevented from occurring, denoted by $\Sigma_{c,i}$(for $i \in I$) and *uncontrollable* events that controller i cannot prevent from occurring, denoted by $\Sigma_{uc,i}$. The overall set of controllable events is $\Sigma_c := \cup_{i=1}^{n} \Sigma_{c,i}$ and the overall set of uncontrollable events is $\Sigma_{uc} := \Sigma \setminus \Sigma_c$. For all $\sigma \in \Sigma_c$, we define $I_c(\sigma) = \{i \in I \mid \sigma \in \Sigma_{c,i}\}$. The specification K is *controllable* wrt L and Σ_{uc} if

$$\overline{K}\Sigma_{uc} \cap L \subseteq \overline{K}. \tag{1}$$

Another aspect of the control problem involves the notion of partial observation: there are some events that controller i can observe, namely $\Sigma_{o,i}$ for $i \in I$, while the rest of the events in Σ are unobservable to controller i, denoted by $\Sigma_{uo,i}$.

To formally capture the notion of partial observation, we define a *canonical projection* $\pi_i : \Sigma^* \rightarrow \Sigma_{o,i}^*$. Thus for $t = \sigma_1\sigma_2 \ldots \sigma_m \in \Sigma^*$, the partial observation $\pi_i(t)$ will contain only those events $\sigma_i \in \Sigma_{o,i}$, since unobservable events are removed. The specification K is *co-observable* [19] wrt L, $\Sigma_{o,i}$, and $\Sigma_{c,i}$ if

$$\begin{aligned}(\forall t \in \overline{K})(\forall \sigma \in \Sigma_c) t\sigma \in L \setminus \overline{K} \Rightarrow \\ (\exists i \in I_c(\sigma)) \pi_i^{-1}[\pi_i(t)]\sigma \cap \overline{K} = \emptyset.\end{aligned} \tag{2}$$

When $I = \{1\}$, K is said to be *observable* [10].

A decentralized control law for controller i is a mapping $\Gamma_i : \pi_i(L) \rightarrow Pwr(\Sigma)$ that defines the set of events that controller i believes should be *enabled* based on its partial view of the system behavior. While controller i can choose to enable or disable elements in $\Sigma_{c,i}$, it must enable all events in $\Sigma_{uc,i}$.

$$(\forall i \in I)(\forall t \in L)\ \Gamma_i(\pi_i(t)) = \{\gamma \in Pwr(\Sigma) \mid \gamma \supseteq \Sigma_{uc,i}\}.$$

Admissible decentralized control laws Γ_i allow local decisions to be taken in an observationally-equivalent fashion:

$$\begin{aligned}(\forall t, t' \in L)(\forall i \in I)\ \pi_i(t) = \pi_i(t') \Rightarrow \\ (\sigma \in \Gamma_i(t) \Rightarrow \sigma \in \Gamma_i(t')).\end{aligned} \tag{3}$$

To find a solution to the decentralized control problem in the absence of communication between controllers, we want to find Γ_i ($\forall i \in I$) such that $\forall t \in \overline{K}$:

$$\begin{aligned}&(\forall \sigma \in (\Sigma_c \vee \Sigma_{uc})) t\sigma \in \overline{K} \Rightarrow \sigma \in \cap_{i\in I}\Gamma_i(\pi_i(t)) \wedge \\ &(\forall \sigma \in \Sigma_c) t\sigma \in L \setminus \overline{K} \Rightarrow \sigma \notin \cap_{i\in I}\Gamma_i(\pi_i(t)).\end{aligned}$$

From the results of [19], such Γ_i ($\forall i \in I$) exist if the specification K is co-observable (wrt L, $\Sigma_{o,i}$ and $\Sigma_{c,i}$), controllable (wrt L and Σ_{uc}), and L_{m}-closed.

When K does not satisfy Eq. (2) and $n \geq 2$, it may still be possible to find a control solution by introducing synchronous communication between controllers. We know from [2], [18] that we can synthesize synchronous communication protocols when K is controllable (wrt L and Σ_{uc}), L_{m}-closed, observable (wrt L, Σ_o and Σ_c) but is *not* co-observable (wrt L, $\Sigma_{o,i}$ and $\Sigma_{c,i}$). From now on in this paper, we assume that K is controllable, observable and L_{m}-closed.

The synthesis of communication protocols requires the introduction of a set of messages Δ that controllers send to each other. Let $\Delta = \bigcup_{\substack{i,j\in I \\ i\neq j}} \Delta_{i,j}$, where $a \in \Delta_{i,j}$ is a message that controller i sends to controller j. For the problem that we consider here, $\Delta_{i,j} \subseteq \Sigma_{o,i} \setminus \Sigma_{o,j}$. It could be the case that no message is sent, in which case the controller is silent (ε). Let $\Delta_{i,j}^{\varepsilon} := \Delta_{i,j} \cup \{\varepsilon\}$. A *synchronous communication protocol* between controllers $i, j \in I$ is a mapping $\phi_{i,j} : L \rightarrow \Delta_{i,j}^{\varepsilon}$ and indicates the message that is synchronously sent from controller i to controller j.

The latest information that a controller can keep through a sequence can be defined as $\psi_i : L \rightarrow \Delta_{i,j}^{\varepsilon} \cup (\cup_{\substack{j\in I \\ i\neq j}} \Delta_{j,i})$, such that when $t = \sigma_1 \ldots \sigma_m \in L$ occurs, each controller i keeps track of communication it receives about t along with its own observations of t.

$$\psi_i(t) = \begin{cases} \sigma_m, & \text{if } \sigma_m \in \Sigma_{o,i} \text{ or} \\ & \sigma_m \notin \Sigma_{o,i} \text{ and } \exists j \in I \text{ s.t. } \phi_{j,i}(t) \neq \varepsilon; \\ \varepsilon, & \text{otherwise.} \end{cases}$$

The canonical projection π is extended to include received messages: $\pi_i^{\Delta} : \Sigma^* \rightarrow (\Sigma_{o,i} \cup (\cup_{\substack{j\in I \\ i\neq j}} \Delta_{j,i}))^*$, where $\pi_i^{\Delta}(\varepsilon) = \varepsilon$, and $\pi_i^{\Delta}(t) = \psi_i(\sigma_1)\, \psi_i(\sigma_1\sigma_2) \ldots \psi_i(\sigma_1 \ldots \sigma_m)$, for $t = \sigma_1 \ldots \sigma_m$.

Finally, it must be the case that communication occurs in an observationally-equivalent manner. Communication protocols $\phi_{i,j}$ are admissible if

$$\begin{aligned}(\forall t, t' \in L)(\forall i \in I)\ \pi_i^{\Delta}(t) = \pi_i^{\Delta}(t') \Rightarrow \\ (\forall j \in I \setminus \{i\})\ \phi_{i,j}(t) = \phi_{i,j}(t').\end{aligned} \tag{4}$$

We extend the decentralized control law to a communicating controller i as follows $\Gamma_i^{\Delta} : \pi_i^{\Delta}(L) \rightarrow Pwr(\Sigma)$. To find a solution to the decentralized control problem with synchronous communication protocols $\Phi = \{\phi_{i,j}\}$ for all controllers $i, j \in I$, we have to find Γ_i^{Δ} ($\forall i \in I$) such that

$\forall t \in \overline{K}$:

$$(\forall \sigma \in (\Sigma_c \vee \Sigma_{uc}))t\sigma \in \overline{K} \Rightarrow \sigma \in \cap_{i\in I}\Gamma_i^{\Delta}(\pi_i^{\Delta}(t)) \wedge$$
$$(\forall \sigma \in \Sigma_c)t\sigma \in L \setminus \overline{K} \Rightarrow \sigma \notin \cap_{i\in I}\Gamma_i^{\Delta}(\pi_i^{\Delta}(t)). \quad (5)$$

From the results of [17], we can find such Γ_i when K is co-observable with respect to L, π_i^{Δ}, and $\Sigma_{c,i}$, controllable (wrt L and Σ_{uc}), and L_{m}-closed.

III. MULTIOBJECTIVE OPTIMIZATION OF DECENTRALIZED DES WITH COMMUNICATION

A multiobjective optimization problem is characterized by the requirement to optimize multiple conflicting objectives. Evolutionary algorithms are used for solving multiobjective optimization problems. The idea of such algorithms is as follows: beginning with an initial population of possible solutions, each solution is assigned a *fitness* value indicating its quality. The fitness value determines which solutions will be selected for breeding the next *generation*. These candidates are mutated and combined to produce new "children" candidate solutions. The evolutionary process continues until either an optimal set of solutions is determined or a predetermined number of generations is exceeded.

There may not exist a single best solution in the multiobjective optimization problem. Instead, evolutionary algorithms define a *set* of best solutions. The class of evolutionary algorithms that we are using produces a *Pareto front* of the candidate solutions. Solutions that comprise the front are said to be *Pareto-optimal* or *non-dominated*. A solution x_1 is said to be dominated by another solution x_2, if x_1 is not better than x_2 in any objectives, and x_1 is strictly worse than x_2 in at least one objective. Most evolutionary multiobjective optimization approaches such as strength Pareto evolutionary algorithm (SPEA) [25], non-dominated sorting genetic algorithms (NSGA-II) [5], and the Pareto-archived evolution strategy (PAES) [8] use the concept of domination. To solve our multiobjective DES problem, we use NSGA-II. Unlike some of the other approaches, NSGA-II keeps an archive of the best b solutions generated so far: all children of generation k compete for membership in generation $k+1$ with generation k. In this way, good solutions from a previous generation are preserved. The algorithm also features a strong fitness assignment procedure.

For decentralized DES with communication, we have two objectives to optimize: each decentralized controller i must optimize the cost of its local control law v_i and the cost of its local communication policy u_i. Ideally, we would like the joint decisions of the controllers in the presence of the full communication protocol to allow exactly K to occur; however, in the presence of a costly communication protocol, it might be more efficient to allow some subset of K to occur. But it may be the case that the penalty for disabling certain sequences within K is more than the communication required to enable the same sequence. We are interested in a quantitative analysis of the trade-off between the cost of imperfectly controlling the system by removing some (potentially costly) communications and the cost of taking exact control solution with the full communication protocol.

We adapt the centralized control cost function of [21] to the case of the control cost function for a decentralized controller i (for $i \in I$). We consider three basic costs that controller i (for $i \in I$) can incur to control a system:

1) We assume that there is a basic cost for an event to occur, which can be considered to be the cost to enable an event , denoted by $e_i : \Sigma \to \mathbb{R}^+ \cup \{0\}$.
2) There is a cost to disable an event that would otherwise take the system out of K, $d_i : \Sigma \to \mathbb{R}^+ \cup \{0, \infty\}$.
3) Since our control objective is to have the collection of Γ_i (for $i \in I$) allow exactly K to occur, when a transition is disabled that would otherwise keep the system in K, the cost to disable is incurred, plus an additional penalty is assessed: $p_i^K \in \mathbb{R}$.

We assume that a disablement (and any associated penalty) or an enablement cost lies in the range of $[0, \infty)$. When a controller tries to disable an uncontrollable event, a penalty of ∞ is levied. When a controller is not sure whether or not the system leaves K via a controllable event, corresponding to an "uncertain" decision for controller i, the default decision is to enable the event. We consider the cost of an uncertain control decision to be cost of enablement. Because we will consider only control laws that keep the system within K, it will not be possible that all controllers enable same event that takes the system out of K. Note that the costs considered here are associated with a controller's local decision regarding the occurrence of an event, and not for the eventual fusion of the control decisions.

The control cost $v_i : \Gamma_i^{\Delta} \times \overline{K} \times \Sigma \to \mathbb{R}^+ \cup \{0, \infty\}$ describes the cost incurred by controller i for the occurrence of event $\sigma \in \Sigma$ for $t \in \overline{K}$ such that $\delta(q_0, t\sigma)!$:

$$v_i(\Gamma_i^{\Delta}, t, \sigma) = \begin{cases} e_i(\sigma), & \text{if } \sigma \in \Gamma_i^{\Delta}(\pi_i^{\Delta}(t)); \\ d_i(\sigma), & \text{if } \sigma \notin \Gamma_i^{\Delta}(\pi_i^{\Delta}(t)) \text{ and} \\ & \pi_i^{\Delta^{-1}}[\pi_i^{\Delta}(t)]\sigma \cap \bar{K} = \emptyset; \\ d_i(\sigma) + p_i^K, & \text{if } \sigma \notin \Gamma_i^{\Delta}(\pi_i^{\Delta}(t)) \text{ and} \\ & \pi_i^{\Delta^{-1}}[\pi_i^{\Delta}(t)]\sigma \cap \bar{K} \subseteq \bar{K}; \\ \infty, & \text{otherwise.} \end{cases} \quad (6)$$

The total control cost for controller i (for $i \in I$) is then $V_i(\Gamma_i^{\Delta}, \overline{K}, \Sigma) = \sum_{t\in\overline{K}} \sum_{\sigma\in\Sigma} v_i(\Gamma_i^{\Delta}, t, \sigma)$.

Each decentralized controller i has a communication protocol $\Phi_i = \langle \phi_{i,1}, \ldots, \phi_{i,j}, \ldots, \phi_{i,n} \rangle$. We assume that a basic cost for communication is incurred each time controller i sends a message to controller j, denoted by $\mathsf{com}_i : \Sigma \to \mathbb{R}^+ \cup \{0\}$. The cost of controller i's communication protocol $u_i : \Phi_i \times \overline{K} \times \Sigma \to \mathbb{R}^+ \cup \{0\}$ assumes that a cost is incurred only when a communication is sent by controller i:

$$u_i(\Phi_i, t, \sigma) = \begin{cases} \mathsf{com}_i(\sigma), & \text{if } (\exists j \in I)\ \phi_{i,j}(\pi_i^{\Delta}(t)) = \sigma; \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

It is possible that two identical messages sent by different controllers incur different local costs. There is no cost for the reception of a message and it may be the case that the cost for a point-to-point communication differs from that of a broadcast. For simplicity, we assume that when controller i communicates the same message to more than one controller, a single cost is incurred, regardless of the number of recipients.

The communication cost for all $t \in \overline{K}$ for controller i (for $i \in I$) is then $U_i(\Phi_i, \overline{K}, \Sigma) = \sum_{t \in \overline{K}} \sum_{\sigma \in \Sigma} u_i(\Phi_i, t, \sigma)$. Note that when considering the costs for control and communication in cyclic systems, the total cost function should be updated accordingly to those calculating average cost [17] over an infinite horizon.

To ensure that the objective functions are defined across the same domain, we adjust the definition of U_i and V_i accordingly, so that both functions are defined over $\Gamma_i^{\Delta} \times \Phi_i \times \overline{K} \times \Sigma$.

Objective 1: The first objective function is the cost of the control decisions each controller $i \in I$ makes for its observation of $\overline{K}$:

$$O_{1,i}(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma) = V_i(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma). \tag{8}$$

Objective 2: The second objective function is the cost of the communication protocol that each controller uses to assist the other controllers in reaching the control objective for $\overline{K}$:

$$O_{2,i}(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma) = U_i(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma). \tag{9}$$

We consider an optimization problem with a finite set of control laws Γ^{Δ}, and a finite set of communication protocols Φ. We want to find the trade-off in minimizing the cost of imposing a costly communication protocol Φ on the uncontrolled system L to reach our control objective as compared to eliminating some of the communication and taking a penalty for not reaching the control objective.

Problem 1: Given $K \subseteq L$, find Γ_i^{Δ} and Φ_i ($\forall i \in I$) to

$$\min_{\Gamma_i^{\Delta} \times \Phi_i} \quad f_i(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma) = [O_{1,i}(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma), \\ O_{2,i}(\Gamma_i^{\Delta}, \Phi_i, \overline{K}, \Sigma)]^T,$$

subject to $\emptyset \subset \cap_{i=1}^{n} \Gamma_i^{\Delta}(\pi_i^{\Delta}(\overline{K})) \subseteq \overline{K}$, and $\Gamma^{\Delta} = \langle \Gamma_1^{\Delta}, \ldots, \Gamma_n^{\Delta} \rangle$, $\Phi = \langle \Phi_1, \ldots, \Phi_n \rangle$ are admissible.

We address Problem 1 by applying the evolutionary algorithm NSGA-II [5]. The main algorithms required to describe NSGA-II are presented in [11]. We create an initial population of pairs of possible control laws and communication protocols $\langle \Gamma_i^{\Delta}, \Phi_i \rangle$ that satisfy the constraints of Problem 1. In accordance with NSGA-II, each member of the population is assigned a fitness value, calculated wrt the values of the two objective functions. From the initial population, candidate members for the Pareto front are calculated: those members of the population that are non-dominated. The next generation is calculated following a "breeding" process of elements from the preceding generation. Admissibility of potential control and communication solutions is determined during breeding. Those members of the previous and current population with the best fitness values are then ranked and reorganized into a new candidate set for the Pareto front. This process continues until either we exceed the number of pre-specified generations or the ideal Pareto front is found.

Note that at the conclusion of the algorithm, we have a set of optimal solutions from which to choose. In particular, solutions to Problem 1 provide Pareto optimal costs with respect to communicating controller i. Thus, the designer is free to choose a solution that favours one controller over another, based on the Pareto-fronts produced for each controller. One possible strategy to arrive at a set of global solutions for the locally optimal possibilities from the DES adaptation of NSGA-II is the *Hierarchization Algorithm* from [6].

A. Example:

Let us consider a problem in the space science where a number of robots navigate to explore an area of a planet. The area map is divided into square boxes, where the robots can move from one box to another, either horizontally (left-right), or vertically (up-down). Each movement is represented by an event, and each event occurs at a cost. The event cost in one direction may be higher than the other direction, e.g., if the surface is steep in one direction, then the robots need more energy to move than the other direction. In general, we can divide the area into $m \times m$ square boxes. Suppose there are n robots to explore the area, and more than n target states where the robots want to reach. Their actions are subject to a single constraint: no two robots can occupy the same target state at any time.

For simplicity, in this example, we consider a 3×3 map (m = 3) and n = 2 robots, denoted by R_1 and R_2, each having 2 target states to reach. The automaton for each robot is shown in Fig. 1. An event $xyi \in \Sigma_i$ corresponds to a move from state x to state y by R_i. All events are locally controllable (e.g., R_i controls only events that end in i). Similarly, all events are locally observable (e.g., R_i observes only events that end in i). R_1 starts from state 1 and has target states are 7 and 8 whereas R_2 starts from state 3 and has target states 8 and 9. According to the constraint noted above, R_1 and R_2 cannot be in state 8 at the same time.

The system behavior L is the language generated by the synchronous product of $\mathsf{R}_1||\mathsf{R}_2$. The corresponding automaton M_L has 81 states and 234 transitions[1]. The specification automaton M_K is a subautomaton of M_L, missing only (8,8) from M_L and the transitions associated with that state.

The robots have a map of the area, but no robot knows in which target state it will end up. For instance, if R_1 reaches state 7, then R_2 can go to either state 8 or state 9. But if R_1 has already reached state 8, then R_2 must be informed about the position of R_1, so that R_2 can move to state 9. In fact, to avoid the situation when both R_1 and R_2 are in state 8, it is necessary that each robot inform the other whenever the following events occur: $\mathsf{58}i$, $\mathsf{78}i$ or $\mathsf{98}i$, for $i \in \{1, 2\}$.

[1]We used the software DESUMA to calculate the synchronous product.

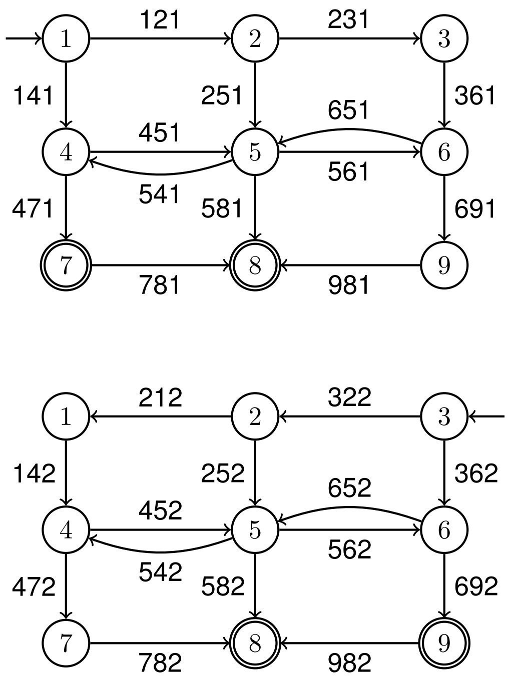

Fig. 1: The automaton model for (a) R_1; (b) R_2.

TABLE I: Non-dominated solutions of Controller 1.

$u_1^*(\cdot)$	$v_1^*(\cdot)$	$u_2(\cdot)$	$v_2(\cdot)$
0	37,700	106,000	291,200
2300	26,700	105,800	1,306,300
1000	27,450	87,200	2,321,500
3300	26,150	107,200	1,294,500
12800	21,200	86,000	291,200

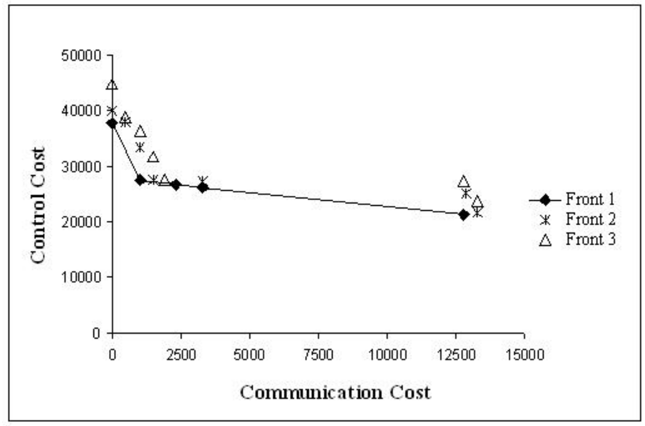

Fig. 2: Pareto fronts of rank 1,2,3 for Controller 1 after 100 generations.

Three basic costs are assigned for the control cost function:

$$e_1(\sigma) = \begin{cases} 50, & \text{if } \sigma \in \{121,141,231,251,361\}; \\ 100, & \text{if } \sigma \in \{451,471,541,561,581,651,691\}; \\ 150, & \text{if } \sigma \in \{781,981\}. \end{cases}$$

$$e_2(\sigma) = \begin{cases} 100, & \text{if } \sigma \in \{142,212,252,322,362\}; \\ 150, & \text{if } \sigma \in \{452,472,542,562,582,692\}; \\ 200, & \text{if } \sigma \in \{782,982\}. \end{cases}$$

$$d_1(\sigma) = \begin{cases} 2000, & \text{if } \sigma \in \{121,141,231,251,361\}; \\ 1500, & \text{if } \sigma \in \{451,471,541,561,581,651,691\}; \\ 1000, & \text{if } \sigma \in \{781,981\}. \end{cases}$$

$$d_2(\sigma) = \begin{cases} 1500, & \text{if } \sigma \in \{142,212,252,322,362\}; \\ 1000, & \text{if } \sigma \in \{452,472,542,562,582,692\}; \\ 500, & \text{if } \sigma \in \{782,982\}. \end{cases}$$

and

$$p_1^K = 3000,$$
$$p_2^K = 4000.$$

The cost of a communication is defined as below.

$$\mathsf{com}_1(\sigma) = \begin{cases} 500, & \text{if } \sigma \in \{581\}; \\ 10000, & \text{if } \sigma \in \{781\}; \\ 900, & \text{if } \sigma \in \{981\}; \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathsf{com}_2(\sigma) = \begin{cases} 500, & \text{if } \sigma \in \{582\}; \\ 700, & \text{if } \sigma \in \{782\}; \\ 20000, & \text{if } \sigma \in \{982\}; \\ 0, & \text{otherwise.} \end{cases}$$

We illustrate Algorithm NSGA-II for $\mathsf{R}_1||\mathsf{R}_2$. The initial size of the population, $|P|$ is 40 and the algorithm was run for 100 generations. The first three ranks of the Pareto front for Controller 1 are shown in Fig. 2. The non-dominated solutions for Controller 1 (the solutions in the front of rank 1) are shown in Table I, which represents the best compromises for Robot 1. In particular, the five Pareto-optimal solutions offer a variety of possible costs for a communication policy for Controller 1, ranging from a cost of 0 up to a cost of 12,800. It is interesting to note that when Controller 1 does not communicate anything to Controller 2, the resulting solution for Controller 2 incurs a communication cost of 106,000. Whereas when Controller 1 increases its communication so that the cost of communicating Φ_1 is 1000, the communication cost for Controller 2 decreases to 87,200, but the control cost increases nearly eight-fold.

The first three ranks of the Pareto front for Controller 2 are shown in Fig. 3. The non-dominated solutions for Controller 2 (the solutions in the front of rank 1) are shown in Table II, which are the best compromises for Robot 2. Again, it is interesting to examine the Pareto-optimal solutions for Controller 2: when Controller 2 does not communicate at all, the control cost for Controller 1 is 1,265,700. Whereas when Controller 2 communicates a bit more with a cost of 500,

TABLE II: Non-dominated solutions of Controller 2.

$u_1(\cdot)$	$v_1(\cdot)$	$u_2^*(\cdot)$	$v_2^*(\cdot)$
55,100	1,265,700	0	42,050
46,000	1,280,300	1900	35,400
26,100	1,325,800	2400	29,000
54,300	1,248,650	500	36,000
56,500	308,500	21,900	25,250

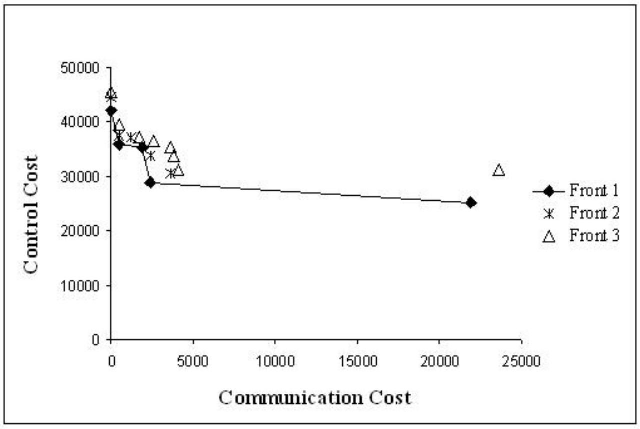

Fig. 3: Pareto fronts of rank 1,2,3 for Controller 2 after 100 generations.

the control cost for Controller 1 only goes down by about 1.3%.

Ultimately, we use NSGA-II as an initial guide to aid in the selection of local Pareto-optimal communication and control policies for decentralized DES. For instance, there may be compelling physical arguments to insist that one decentralized site assumes the bulk of the communication during the operation of system tasks, despite the site incurring a high communication cost (wrt other sites). Similarly, we may be willing for some degree of approximation on one or more sites to reduce the cost of communication to achieve a precise control decision. Modeling the trade-off as a multioptimization problem gives us a better selection of optimal solutions from which to chose communication and control policies for this class of DES problem.

ACKNOWLEDGMENTS

The authors would like to thank Mahmudul Hasan for implementing the algorithm in Visual C++.

REFERENCES

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms,* Oxford University Press, 1996.

[2] G. Barrett and S. Lafortune, "Decentralized Supervisory Control with Communicating Controllers," *IEEE Trans. Automat. Contr.*, vol. 45, no. 9, pp.1620–38, 2000.

[3] K. Chatterjee, L. Doyen and T.A. Henzinger,"Quantitative languages," *ACM Trans. Comput. Logic*, vol. 11, no. 4, 2010.

[4] I. Chattopadhyay and A. Ray, "A Language Measure for Partially Observed Discrete Event Systems," *International journal of control*, vol. 79, no. 9, pp. 1074–1086, 2006.

[5] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA - II," *IEEE Trans. Evolut. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[6] E. Dumitrescu, A. Girault, H. Marchand and E. Rutten, "Multicriteria Optimal Reconfiguration of Fault-Tolerant Real-Time Tasks,", In *Proc. WODES,* Berlin, Germany, 2010, pp. 366–373.

[7] A. Gambier and E. Badreddin, "Multi-objective Optimal Control: An Overview", In *Proc.16th IEEE International Conference on Control Applications*, Singapore, 2007, pp. 170–175.

[8] J. Knowles, and D. Corne,"The Pareto Archived Evolution Strategy: A new Baseline Algorithm for Multiobjective Optimization," *Proc. the 1999 Congress on Evolutionary Computation*, New Jersey, 1999, pp. 98–105.

[9] R. Kumar and V.K. Garg, "Optimal Supervisory Control of Discrete Event Dynamical Systems," *SIAM J. Control and Optim.*, vol. 33, 1995, pp. 419–439.

[10] F. Lin and W.M. Wonham, "On Observability of Discrete-Event Systems," *Information Sciences*, vol. 44, no. 3, pp. 173–198, 1988.

[11] S. Luke, *Essentials of Metaheuristics*, Lulu, 2009. Available at http://cs.gmu.edu/∼sean/book/metaheuristics/

[12] H. Marchand, O. Boivineau and S. Lafortune, "On Optimal Control of a Class of Partially-Observed Discrete Event Systems," *Automatica*, vol. 38, no. 11, 2002, pp. 1935–1943.

[13] A. Overkamp and J.H. van Schuppen, "Maximal Solutions in Decentralized Supervisory Control," *SIAM J. Control Optim.*, vol. 39, no. 2, pp. 492–511, 2000.

[14] R. Porter, E. Nudelman and Y. Shoham, "Simple Search Methods for Finding a Nash Equilibrium," *Games and Economic Behavior*, vol. 63, no. 2, pp. 642–662, 2008.

[15] P.J. Ramadge and W.M. Wonham, "Supervisory Control of a Class of Discrete Event Processes," *SIAM J. Contr. Optim.*, vol. 25, no.1, pp. 206–230, 1987.

[16] A. Ray, A. Surana and S. Phoha, "A Language Measure for Supervisory Control," *Applied Mathematics Letters*, Vol. 16, September 2003, pp. 985–991.

[17] S.L. Ricker, "Asymptotic Minimal Communication for Decentralized Discrete-Event Control", *Proc. WODES*, Goteburg, Sweden, 2008, pp. 486–491.

[18] S.L. Ricker and B. Caillaud, "Mind the Gap: Expanding Communication Options in Decentralized Discrete-Event Control," in *Proc. IEEE Conf. Decision Contr.*, New Orleans, LA, 2007, pp. 5924–5929.

[19] K. Rudie and W.M. Wonham,"Think Globally, Act Locally: Decentralized Supervisory Control," *IEEE Trans. Automat. Contr.*, vol. 37, no. 11, pp. 1692–1708, Nov. 1992.

[20] W.H. Sadid, S.L. Ricker and S.Hashtrudi-Zad,"Nash Equilibrium for Communication Protocols in Decentralized Discrete-Event Systems," *Proc. ACC*, Baltimore, 2010, pp. 3384–3389.

[21] R. Sengupta and S. Lafortune, "An Optimal Control Theory for Discrete Event Systems," *SIAM J. Contr. Optim.*, vol. 36, no. 2, pp. 488–541, 1998.

[22] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley, New York, 1986.

[23] D. P. L. Thorsley, "Application of Stochastic Techniques to Partially Observed Discrete Event Systems", Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Michigan, 2006.

[24] P. Woźniak, "Multi-objective Control Systems Design with Criteria Reduction", LNCS 6457, 2010, pp. 583–587.

[25] E. Zitzler, K. Deb and L. Thiele,"Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.