2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10885812

Compositional Planning for Logically Constrained Multi-Agent Markov Decision Processes

Krishna C. Kalagarla*[1,2], Matthew Low*[1], Rahul Jain[1], Ashutosh Nayyar[1], Pierluigi Nuzzo[1,3]
[1]Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles
[2]Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque
[3]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA
Email: {kalagarl,lowmatth,rahul.jain,ashutosn,nuzzo}@{[1]usc.edu, [2]unm.edu, [3]eecs.berkeley.edu}

Abstract— **Designing control policies for large, distributed systems is challenging, especially in the context of critical, temporal logic based specifications (e.g., safety) that must be met with high probability. Compositional methods for such problems are needed for scalability, yet relying on worst-case assumptions for decomposition tends to be overly conservative. In this work, we use the framework of Constrained Markov Decision Processes (CMDPs) to provide an assume-guarantee based decomposition for synthesizing decentralized control policies, subject to logical constraints in a multi-agent setting. The returned policies are guaranteed to satisfy the constraints with high probability and provide a lower bound on the achieved objective reward. We empirically find the returned policies to achieve near-optimal rewards while enjoying an order of magnitude reduction in problem size and execution time.**

I. Introduction

Our increasingly connected and "smart" world calls for compositional methods to design control policies for large, distributed systems in a scalable manner. Smart power grids and intersections are applications in which a single, centralized approach is not possible [1], [2]. Moreover, these are critical systems, which must be carefully controlled to realize their intended behavior. Constrained Markov decision processes (CMDPs) [3], [4] are a powerful mathematical model for representing sequential decision-making tasks subject to certain constraints under uncertainty, making them a viable choice to model such systems. CMDPs can be infused with mission specifications, expressed in logic languages such as finite linear temporal logic (LTL_f) [5], [6], [7], to ensure that returned policies respect and achieve the mission to a user-specified probability threshold. However, centralized approaches to solve CMDPs with multiple agents suffer from the combinatorial explosion of the global state space. Directly solving the monolithic CMDP entails solving an expanding constrained optimization problem, which, even for two agents, quickly becomes untenable for larger problems.

Approaches to decompose the monolithic optimization problem into more manageable pieces are of significant interest [8]. Worst-case or robust control decomposes the problem in an adversarial manner, where each agent assumes the worst-case behavior of the other agents with respect to (w.r.t.) some objective function [9], [10]. The opposite is to optimistically assume cooperation among agents to decompose the joint optimization problem [11]. These methods are most commonly seen in the context of multi-agent, unconstrained optimal control and reinforcement learning [12]. Compositional methods capable of handling logical specifications, however, have received scant attention. To the best of our knowledge, this paper is the first to provide a compositional strategy to solve logically constrained Markov decision processes (MDPs) in the multi-agent setting.

Neither worst-case nor pure optimism are decomposition paradigms reflective of real-world interactions between agents, as perfect cooperation is often unrealizable and fully adversarial methods tend to be overly conservative. Instead, our approach takes a middle-ground approach inspired by assume-guarantee (AG) reasoning and contract-based design [13], [14]. Consider the scenario for a pair of agents. In our framework, each agent *assumes* that the other will obey its corresponding logical constraints with some high probability. Under this assumption, the ego agent finds an optimal policy by considering worst-case behavior of its partner w.r.t. the joint objective reward, subject to the before mentioned constraint. The returned policies are *guaranteed* to satisfy the ego agent's logical constraints with high probability and provide a lower bound on the achieved joint reward. Mutual understanding of undesirable outcomes enables efficient synthesis of provably safe, optimal policies with an empirically tight optimality gap.

In this paper, we (1) introduce a novel, AG-based decomposition of the monolithic CMDP formulation, (2) show how this formulation can be efficiently transformed and solved as a linear program (LP), and (3) validate our methodology on two case studies to demonstrate the computational advantages of our modular optimal policy synthesis approach while ensuring provable logical constraint satisfaction.

II. Preliminaries

Notation: Real and natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}$, respectively. General probabilities are specified by $\mathbb{P}$, while transition probability functions use P. We use $h \in [i : j]$ (where $[i : j]$ is the inclusive sequence of integers from i to j) to denote a time step inside an episode. The indicator function $\mathbb{1}_{s_1}(s)$ evaluates to 1 when $s = s_1$ and 0 otherwise. The probability simplex over the set S is denoted by Δ_S. For a string s, $|s|$ denotes the length of the string. The Cartesian product over sets is defined by $\times$, while $\cdot$ is used for standard multiplication. Superscripts on MDP elements denote the agent index and product status, while subscripts denote the current time step.

*Equal contribution. The authors wish to acknowledge the partial support of the National Science Foundation under awards CNS 1846524, ECCS 2139982, ECCS 2025732, and ECCS 1750041.

A. Labeled Finite-Horizon MDPs

We consider labeled finite-horizon MDPs [3], formally defined by a tuple $\mathscr{M} = (S, A, H, s_1, P, r, AP, L)$, where S and A denote the finite state and action spaces, respectively. The agent interacts with the environment in episodes of length H, with each episode starting from the same initial state s_1. The non-stationary transition probability is P, where $P_h(s'|s,a)$ is the probability of transitioning to state s' upon taking action a in state s at time step $h \in [1:H]$. The deterministic, non-stationary reward of taking action a in state s at time step h is $r_h(s,a)$. AP is a set of atomic propositions, e.g., indicators of the truth value for the presence of an obstacle or goal. $L : S \to 2^{AP}$ is a labeling function which indicates the set of atomic propositions which hold true in each state, e.g., $L(s) = \{y\}$ indicates that only the atomic proposition y is true in state s.

A non-stationary randomized policy $\pi = (\pi_1, \dots, \pi_H) \in \Pi$, where $\pi_i : S \to \Delta_A$, maps each state to a probability distribution over the action space. A *run* ξ of the MDP is the sequence of states and actions $(s_1, a_1) \dots (s_H, a_H)$. The total expected reward of an episode associated with a policy π and reward function r is given by

$$\mathcal{R}_\pi^{\mathscr{M}}(r) = \mathbb{E}_\pi^{\mathscr{M}} \left[\sum_{i=1}^{H} r_i(s_i, a_i) \right]. \tag{1}$$

In this paper, we will make use of constrained MDPs (CMDPs) [4], which additionally include a constraint reward function $c_h(s,a)$ at each time step h. The total expected constraint reward in an episode under a policy π is defined in the same manner as (1) with r_h replaced by c_h. The goal of the CMDP problem is to find a policy π^* that maximizes the objective total reward $\mathcal{R}_\pi^{\mathscr{M}}(r)$ while ensuring that the total constraint reward is above a threshold l, i.e.,

$$\begin{aligned} \pi^* = \operatorname*{argmax}_{\pi \in \Pi} \quad & \mathcal{R}_\pi^{\mathscr{M}}(r) \\ \text{s.t.} \quad & \mathcal{R}_\pi^{\mathscr{M}}(c) \geq l. \end{aligned} \tag{2}$$

B. Occupancy Measures

Occupancy measures [4], [15] allow for an alternative representation of the set of non-stationary, randomized policies and the expected return of such policies. CMDPs can be solved in terms of occupancy measures, as they enable the search for an optimal policy (2) to be rewritten as a linear program (LP). The occupancy measure q^π of a policy π in a finite-horizon MDP is defined as the expected number of visits to a state-action pair (s,a) in an episode at time step h. Formally, $q_h^\pi(s,a) = \mathbb{P}[S_h = s, A_h = a | S_1 = s_1, \pi]$.

The occupancy measure q^π of a policy π satisfies linear constraints expressing non-negativity, the conservation of probability flow through the states, and the initial state conditions. The space of the occupancy measures satisfying these constraints is denoted by $\mathbb{Q}_{\mathscr{M}}$ and is convex [4]. A policy π generates an occupancy measure $q \in \mathbb{Q}_{\mathscr{M}}$ if

$$\pi_h(a|s) = \frac{q_h(s,a)}{\sum_b q_h(s,b)}, \quad \forall (s,a,h). \tag{3}$$

Thus, there exists a non-stationary, randomized policy for each occupancy measure in $\mathbb{Q}_{\mathscr{M}}$ and *vice versa*. Further, the total expected reward of an episode under policy π with respect to reward function r can be expressed in terms of the occupancy measure as $\mathcal{R}_\pi^{\mathscr{M}}(r) = \sum_{h,s,a} q_h^\pi(s,a) r_h(s,a)$.

C. Finite Linear Temporal Logic Specification

We use LTL_f [5], a temporal extension of propositional logic. This is a variant of linear temporal logic (LTL) [16] interpreted over finite traces. LTL_f is flexible enough to express complex finite-duration task specifications, while remaining unambiguous and computer readable. These traits make it an attractive candidate for incorporation with reward functions in a specify-then-synthesize design paradigm [17], [18]. Given a set AP of atomic propositions, LTL_f formulae are constructed inductively as follows:

$$\varphi := \mathsf{true} \mid a \mid \neg\varphi \mid \varphi^1 \wedge \varphi^2 \mid \mathbf{X}\varphi \mid \varphi^1 \mathbf{U} \varphi^2,$$

where $a \in AP$; φ, φ^1, and φ^2 are LTL_f formulae; $\wedge$ and $\neg$ are the logic conjunction and negation; and $\mathbf{U}$ and $\mathbf{X}$ are the *until* and *next* temporal operators. Additional temporal operators such as *eventually* ($\mathbf{F}$) and *always* ($\mathbf{G}$) are derived as $\mathbf{F}\varphi := \mathsf{true}\mathbf{U}\varphi$ and $\mathbf{G}\varphi := \neg\mathbf{F}\neg\varphi$. Formulae are interpreted over finite-length words $w = w_1 \dots w_{|w|}$, where each letter $w_i \subseteq AP$. When φ is *true* for w at step $i \in [1:|w|]$, we write $w, i \models \varphi$. A formula φ is *true* in w, written $w \models \varphi$, iff $w, 1 \models \varphi$.

Given an MDP $\mathscr{M}$ and an LTL_f formula φ, a run $\xi = (s_1, a_1) \dots (s_H, a_H)$ of the MDP under policy π is said to satisfy φ if the word $w = L(s_1) \dots L(s_H) \in (2^{AP})^H$ generated by the run satisfies φ. The probability that a run of $\mathscr{M}$ satisfies φ under policy π is denoted by $\mathbb{P}_\pi^{\mathscr{M}}(\varphi)$.

D. Deterministic Finite Automaton (DFA)

The language defined by an LTL_f formula, i.e., the set of words satisfying the formula, can be captured by a Deterministic Finite Automaton (DFA) [6]. We denote a DFA by a tuple $\mathscr{A} = (Q, \Sigma, q_0, \delta, F)$, where Q is a finite set of states, Σ is a finite alphabet, $q_0 \in Q$ is an initial state, $\delta : Q \times \Sigma \to Q$ is a transition function, and $F \subseteq Q$ is the set of accepting states. A run $\xi_{\mathscr{A}}$ of $\mathscr{A}$ over a finite word $w = w_1 \dots w_n$ (with $w_i \in \Sigma$) is a sequence of states $q_0 q_1 \dots q_n \in Q^{n+1}$ such that $q_{i+1} = \delta(q_i, w_{i+1})$ for $i = 0, \dots, n-1$. A run $\xi_{\mathscr{A}}$ is accepting if and only if (iff) $q_n \in F$. A word w is accepted by $\mathscr{A}$ iff the run $\xi_{\mathscr{A}}$ of $\mathscr{A}$ on w is accepting. Finally, we say that an LTL_f formula is equivalent to a DFA $\mathscr{A}$ iff the language defined by the formula is the set of words accepted by $\mathscr{A}$. For any LTL_f formula φ over AP, we can construct an equivalent DFA with input alphabet 2^{AP}.

III. Problem Formulation

We first describe the optimal policy synthesis problem under LTL_f constraints for one agent and then present our 2-player problem formulation.

Single Player MDP: Given a labeled finite-horizon MDP $\mathscr{M}$ and an LTL_f specification φ, our objective is to design a policy π that maximizes the total expected reward $\mathcal{R}_\pi^{\mathscr{M}}(r)$ while ensuring that the probability $\mathbb{P}_\pi^{\mathscr{M}}(\varphi)$ of satisfying the specification φ is at least $1 - \delta$. More formally, we would like to solve the following constrained optimization problem:

$$\textbf{LTL}_f\textbf{-MDP:} \quad \begin{aligned} \max_{\pi} \quad & \mathcal{R}_\pi^{\mathscr{M}}(r), \\ \text{s.t.} \quad & \mathbb{P}_\pi^{\mathscr{M}}(\varphi) \geq 1 - \delta. \end{aligned} \tag{P1}$$

2-Player MDP: Extending to the 2-player setting, we consider two MDPs $\mathscr{M}^i = (S^i, A^i, H, s_1^i, P^i, AP^i, L^i)$, for $i \in \{1,2\}$, with independent transition probabilities. The two MDPs are connected by a *joint* reward function $r_h^J : (S^1 \times S^2) \times (A^1 \times A^2) \to \mathbb{R}$, where $r_h^J(s_h^J, a_h^J)$ is the reward of taking joint action $a_h^J = (a_h^1, a_h^2)$ in joint state $s_h^J = (s_h^1, s_h^2)$ at time step h. Atomic propositions AP^i are assumed disjoint without loss of generality.

The objective of the 2-player problem is to design a joint policy π^J that maximizes the total expected objective reward while satisfying the joint specification φ^J with probability at least $1-\delta$. The joint specification is the conjunction of the two single-player specifications: $\varphi^J = \varphi^1 \wedge \varphi^2$, where φ^i is a specification defined over the run of $\mathscr{M}^i$.

$$\textbf{LTL}_f\textbf{-MDP-2-Player:} \quad \max_{\pi} \quad \mathcal{R}_{\pi}^{\mathscr{M}^J}(r^J), \quad \text{s.t.} \quad \mathbb{P}_{\pi^J}^{\mathscr{M}_J}(\varphi^J) \geq 1-\delta. \tag{P2}$$

We use $\mathscr{M}^J$ to denote the joint MDP which incorporates the states, actions, and transitions of the component MDPs $\mathscr{M}^1$ and $\mathscr{M}^2$. Details of this construction follow in Section IV-A.

IV. Solution Approach

We first describe the monolithic approach to solve the joint problem formulation P2. This method combines the two agents to obtain a centralized policy over the joint state-action space. This approach yields an LP by utilizing occupancy measures and product CMDP to join the logically specified DFA with the probabilistic MDP [7]. Our AG-based, decentralized approach follows in Section IV-C.

A. Framing 2-Player MDP as Joint MDP

Two CMDPs $(\mathscr{M}^1, \mathscr{M}^2)$ corresponding to the agents in the 2-player setting can be transformed in a single, joint MDP $\mathscr{M}^J$ by the following procedure. The joint state and action spaces are computed by the Cartesian product of the component state and action spaces, i.e., $S^J = S^1 \times S^2$, $A^J = A^1 \times A^2$. The initial state s_1^J is similarly defined. Leveraging the independent transitions of the two MDPs, the joint transition model can be computed by direct multiplication:

$$\begin{aligned} & P_h\left[s_{h+1}^J | s_h^J, a_h^J\right] \\ & = P_h\left[(s_{h+1}^1, s_{h+1}^2) | (s_h^1, s_h^2), (a_h^1, a_h^2)\right] \\ & = P_h\left[s_{h+1}^1, | s_h^1, a_h^1\right] \cdot P_h\left[s_{h+1}^2, | s_h^2, a_h^2\right]. \end{aligned} \tag{4}$$

The joint labeling function is defined as the union of the component MDP labels as follows:

$$L^J(s^J) = L^J((s^1, s^2)) = L^1(s^1) \cup L^2(s^2). \tag{5}$$

The joint LTL$_f$ specification φ^J is converted into a DFA, enabling the computation of the joint product CMDP $\mathscr{M}^{J\times}$ which encapsulates the joint objective and constraint rewards.

With this joint MDP representing both agents, an optimal policy can be found by applying the single player procedure detailed in Section IV-B.

B. Solution Procedure for a Single Player MDP

Given the labeled finite-horizon MDP $\mathscr{M}$ and a DFA $\mathscr{A}$ capturing the LTL$_f$ formula φ, we construct a constrained product MDP $\mathscr{M}^\times = (S^\times, A^\times, H^\times, s_1^\times, P^\times, r^\times, c^\times)$ which incorporates the transitions of $\mathscr{M}$ and $\mathscr{A}$, the reward function of $\mathscr{M}$, and the acceptance set of $\mathscr{A}$.

In the constrained product MDP $\mathscr{M}^\times$, $S^\times = (S \times Q)$ is the set of states, $A^\times = A$ is the action set, and $s_1^\times = (s_1, q_0)$ is the initial state. The horizon length $H^\times$ is $H+1$. For each $s, s' \in S$; $q, q' \in Q$; and $a \in A$, we define the transition function $P_h^\times((s', q')|(s,q), a)$ at time-step $h \in [1:H]$ as

$$P_h^\times((s', q')|(s,q), a) = \begin{cases} P_h(s'|s,a), & \text{if } q' = \delta(q, L(s)) \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The reward functions are defined as

$$r_h^\times((s,q), a) = \begin{cases} r_h(s,a), & \forall s, q, a, h \in [1:H] \\ 0 & \text{if } h = H+1 \end{cases} \tag{7}$$

$$c_h^\times((s,q), a) = \begin{cases} 1, & \text{if } h = H+1 \text{ and } q \in F \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

We thus define the two total expected reward functions on the product MDP: (i) an expected *objective* reward $\mathcal{R}_\pi^{\mathscr{M}^\times}(r^\times)$ associated with the original MDP $\mathscr{M}$, and (ii) an expected *constraint* reward $\mathcal{R}_\pi^{\mathscr{M}^\times}(c^\times)$ associated with reaching an accepting state in the DFA $\mathscr{A}$. For the constrained product MDP $\mathscr{M}^\times$, we are interested in solving the following constrained optimization problem:

$$\textbf{C-MDP:} \quad \max_{\pi} \quad \mathcal{R}_\pi^{\mathscr{M}^\times}(r^\times) \quad \text{s.t.} \quad \mathcal{R}_\pi^{\mathscr{M}^\times}(c^\times) \geq 1-\delta. \tag{P3}$$

Theorem 1 (Equivalence of Problems (P1) and (P3))**.** *For any policy π, we have*

$$\mathcal{R}_\pi^{\mathscr{M}^\times}(r^\times) = \mathcal{R}_\pi^{\mathscr{M}}(r) \tag{9}$$

$$\mathcal{R}_\pi^{\mathscr{M}^\times}(c^\times) = \mathbb{P}_\pi^{\mathscr{M}}(\varphi). \tag{10}$$

Therefore, a policy π^* is an optimal solution in Problem (P1) *if and only if it is an optimal solution to Problem* (P3)*.*

1) Linear Programming Formulation: As described in Section II-B, the constraints corresponding to the occupancy measure definition are created as (11), (12), (13) below:

$$q_h(s,a) \geq 0 \quad \forall s \in S^\times,\ \forall a \in A^\times,\ \forall h \in [1:H^\times], \tag{11}$$

$$\sum_{a \in A^\times} q_h(s,a) = \sum_{s' \in S^\times, a' \in A^\times} P_{h-1}^\times(s|s', a') q_{h-1}(s', a'), \quad \forall s \in S^\times, \forall h \in [2:H^\times], \tag{12}$$

$$\sum_{a \in A^\times} q_1(s,a) = \mathbb{1}_{s_1^\times}(s), \quad \forall s \in S^\times. \tag{13}$$

Additionally, the constraint reward should achieve the specified threshold, i.e.,

$$\sum_{s \in S^\times, a \in A^\times, h \in [1:H^\times]} c_h^\times(s,a) q_h(s,a) \geq 1-\delta. \tag{14}$$

Finally, the LP to maximize the expected reward becomes

$$q^* = \underset{q}{\operatorname{argmax}} \sum_{s\in S^\times, a\in A^\times, h\in[1:H^\times]} r_h^\times(s,a) q_h(s,a), \quad \text{s.t.} \quad (11),(12),(13),(14) \tag{15}$$

The optimal solution q^* of the above LP can be used to obtain the optimal policy π^* using (3).

C. Assume-Guarantee Transformation

When applied in the 2-player setting, the policies obtained by the approach of Section IV-A are necessarily centralized, where each agent must know the current state of all agents. The dimension of the occupancy measure in $\mathscr{M}^{J\times}$ at each time grows rapidly:

$$N = |S^1| \cdot |S^2| \cdot |A^1| \cdot |A^2| \cdot |Q^J|, \tag{16}$$

resulting in significantly larger problems. To overcome the computational burden and the need for centralization in the joint MDP approach, we introduce an (AG) approach to decompose the 2-player CMDP problem into two, smaller optimization problems. Instead of synthesizing a single joint policy π^J that controls both agents, our approach produces two decentralized policies π^1, π^2 corresponding to each agent. These smaller, modular policies allow each agent to act only on local information for independent operation. Decentralized control tends to confer additional benefits such as reduced latency [19], [20].

Decomposition for decentralized policy synthesis is often done by assuming the worst-case policy for the other agent. However, this approach tends to produce overly conservative policies. The key difference of our approach is the use of logical constraints, in an AG framework, to reduce this conservatism. The ego agent is aware of its own logical constraints as well as those specified on the other agent. By assuming that the other agent will obey its constraints, the size of possible policy choices for the other agent is limited. The understanding of this restriction on the other agent's behavior mitigates the conservatism that typically hinders worst-case decomposition [9].

This semi-cooperative, semi-competing framework naturally captures many realistic scenarios between agents. For example, two cars interacting at an intersection can be modeled in this way. Each agent assumes that the other will obey the traffic laws (i.e., each agent's specification) with some high probability, but each agent or driver selfishly looks to minimize its own commute time (objective reward).

D. Formalization of Assume-Guarantee Decomposition

We describe the AG procedure through the lens of one agent, as the product CMDP $\mathscr{M}^{1\times} = \mathscr{M}^1 \times \mathscr{A}^1$ where $\mathscr{A}^1$ is the DFA corresponding to the specification φ^1. The mirrored procedure can be inferred for the second agent $\mathscr{M}^{2\times}$.

Independent AG Policy Synthesis: The agent $\mathscr{M}^{1\times}$ assumes $\mathscr{M}^{2\times}$ will follow some unknown policy π^2 which satisfies the specification φ^2 with probability at least $1-\delta^2$. We guarantee that the returned policy π^1 for $\mathscr{M}^{1\times}$ satisfies its logical constraint φ^1 with probability at least $1-\delta^1$ by construction, i.e.,

$$\begin{aligned} \text{Assume 1}: \quad & \mathcal{R}_{\pi^2}^{\mathscr{M}^{2\times}}(c^{\mathscr{M}^{2\times}}) \geq 1-\delta^2 \\ \text{Guarantee 1}: \quad & \mathcal{R}_{\pi^1}^{\mathscr{M}^{1\times}}(c^{\mathscr{M}^{1\times}}) \geq 1-\delta^1 \end{aligned} \tag{P4}$$

The second agent takes a symmetric view, i.e.,

$$\begin{aligned} \text{Assume 2}: \quad & \mathcal{R}_{\pi^1}^{\mathscr{M}^{1\times}}(c^{\mathscr{M}^{1\times}}) \geq 1-\delta^1 \\ \text{Guarantee 2}: \quad & \mathcal{R}_{\pi^2}^{\mathscr{M}^{2\times}}(c^{\mathscr{M}^{2\times}}) \geq 1-\delta^2. \end{aligned} \tag{P5}$$

Notice that (P4) and (P5) are consistent, in the sense that Guarantee 1 ensures that Assume 2 is valid and Guarantee 2 ensures that Assume 1 is valid.

Theorem 2 (Soundness of AG Policy Composition)**.** *Let π^1 and π^2 be solutions to* (P4) *and* (P5)*, respectively, with δ^1, δ^2 and δ such that $\delta^1 + \delta^2 \leq \delta$. Then, the joint execution of the independent policies as $\pi = (\pi^1, \pi^2)$ is guaranteed to satisfy the conjoined specification φ^J for the joint CMDP $\mathscr{M}^{J\times}$ with probability at least $1-\delta$, i.e.,*

$$\mathcal{R}_{\pi}^{\mathscr{M}^J}(c^{\mathscr{M}^{J\times}}) \geq 1-\delta. \tag{17}$$

Proof. Recall from Theorem 1 that $\mathcal{R}_{\pi}^{\mathscr{M}^\times}(c^\times) = \mathbb{P}_{\pi}^{\mathscr{M}}(\varphi)$. We find the probability of failure for the joint specification:

$$\begin{aligned} \mathbb{P}_{\pi}^{J\times}(\neg\varphi_J) &= \mathbb{P}_{\pi}^{J\times}(\neg(\varphi^1 \wedge \varphi^2)) = \mathbb{P}_{\pi}^{J\times}(\neg\varphi^1 \vee \neg\varphi^2) \\ &\leq \mathbb{P}_{\pi^1}^{1\times}(\neg\varphi^1) + \mathbb{P}_{\pi^2}^{2\times}(\neg\varphi^2) \leq \delta^1 + \delta^2 \leq \delta. \end{aligned}$$

The joint specification is met w.p. at least δ. □

The construction of the independent, AG policy proceeds as follows. Taking $x_h(s,a)$ and $y_h(s,a)$ to be the occupancy measures corresponding to agents $\mathscr{M}^{1\times}$ and $\mathscr{M}^{2\times}$, the optimization problem can be written as an "adversarial" $\max\min$ formulation. In the outer problem, constraints ensuring the occupancy measure validity are equivalent to (11)–(13) when replacing $q, S^\times, A^\times, H^\times, P^\times,$ and $s_1^\times$, by $x, S^{1\times}, A^{1\times}, H^{1\times}, P^{1\times},$ and $s_1^{1\times}$, respectively. They are written as $o_1, o_2,$ and o_3 for brevity. The constraint enforcing the satisfaction of the logical specification is z_1, and it is found by making the same replacements and additionally replacing $c_h^\times, \delta$ with $c_h^{1\times}, \delta^1$ in (14). Together, these constraints complete the outer optimization to yield

$$\max_x \quad f(x), \quad \text{s.t.} \quad o_1, o_2, o_3, z_1 \tag{18}$$

The objective function $f(x)$ of (18) is found by solving the inner optimization problem. Constraints for the inner problem are similarly formed from (11)–(13) by replacing q with y and by replacing the $\mathscr{M}^\times$ elements with those corresponding to the second agent $\mathscr{M}^{2\times}$, e.g., $S^\times$ with $S^{2\times}$, to yield occupancy measure constraints $o_4, o_5,$ and o_6 and mission constraint z_2. The inner problem becomes:

$$f(x) = \min_y \quad g(x,y), \quad \text{s.t.} \quad o_4, o_5, o_6, z_2 \tag{19}$$

where $g(x,y)$ is

$$g(x,y) = \sum_{\substack{s^1\in S^{1\times}, a^1\in A^{1\times}, \\ s^2\in S^{2\times}, a^2\in A^{2\times}, h\in[1:H^\times]}} r_h^\times(s^1,s^2,a^1,a^2) x_h(s^1,a^1) y_h(s^2,a^2). \tag{20}$$

From the perspective of $\mathscr{M}^{1\times}$, this minimization of the joint objective reward corresponds to the worst-case behavior of $\mathscr{M}^{2\times}$ subject to the constraint that φ^2 is satisfied by $\mathscr{M}^{2\times}$. The inclusion of the of logical constraint z_2 effectively blunts the worst-case by restricting the policy space of agent $\mathscr{M}^{2\times}$.

Solving (18) for $\mathscr{M}^{1\times}$ yields an AG optimal policy π^{1*}, induced by occupancy measure x^*, and an associated optimal value v_x^*. Similarly, solving an analogous problem for $\mathscr{M}^{2\times}$ yields π^{2*}, occupancy measure y^*, and optimal value v_y^*.

Theorem 3 (Lower Bound on the Achieved Objective Reward)**.** *The objective rewards v_x^* and v_y^* returned by the AG optimization problems for $\mathscr{M}^{1\times}$ and $\mathscr{M}^{2\times}$ each provide a lower bound on the joint reward achieved when executing policy $\pi = (\pi^{1*}, \pi^{2*})$.*

The proof, omitted for brevity, follows from the adversarial, $\max\min$ formulation of (18).

E. Policy Synthesis as a Linear Program

The nested $\max\min$ formulation is computationally difficult to solve [21]. This challenge is overcome by computing the Lagrangian dual [22] of the inner $\min$ LP, allowing the adversarial problem (18) to be rewritten as a *single* maximization. We obtain the dual of the inner minimization (19) in terms of the new dual variables $\lambda_h^1(s), \lambda^2(s)$, and λ^3 corresponding to the constraints o_5, o_6, and z_2 of the primal problem, respectively. The dual constraints are ℓ_1, ℓ_2, and ℓ_3. Putting together the original outer problem constraints, the dualized inner constraints, and the dualized inner objective, the final LP formulation to find the optimal, independent AG policy for agent $\mathscr{M}^1$ becomes (21).

$$\max_{x,\lambda^3\geq 0,\lambda^1,\lambda^2} \quad \lambda^2(s_1^{2\times}) + (1-\delta^2)\lambda^3, \qquad (21)$$
$$\text{s.t.} \quad o_1, o_2, o_3, z_1$$
$$\ell_1, \ell_2, \ell_3$$

F. Linear Program Size Comparison

We compare the scaling of the LP size for the AG approach against the monolithic construction. Because two optimization problems are solved in the AG framework, the number of variables scales with the larger of the product state-action spaces of the two agents. Assuming that $\mathscr{M}^{1\times}$ has the larger state-action space, the order of the number of LP variables is given below for each approach.

$$\text{AG: } H \cdot |S^1| \cdot |A^1| \cdot |Q^1|$$
$$\text{Monolithic: } H \cdot |S^1| \cdot |S^2| \cdot |A^1| \cdot |A^2| \cdot |Q^J|$$

In the worst case, the maximum size of $|Q^J|$ is given by $|Q^1| \cdot |Q^2|$ [23]. The AG approach has the same number of constraints as variables. In the monolithic case, the number of constraints does not depend on the action space, so the constraints scale with $H \cdot |S^1| \cdot |S^2| \cdot |Q^J|$.

V. Experimental Results

We evaluate our compositional solution to the joint problem against the monolithic approach of Section IV-B in two experiments. Our solution achieves near-optimal objective rewards while improving the execution time by an order of magnitude and maintaining guarantees on constraint satisfaction. The speedup results from avoiding the combinatorial explosion of considering multiple agents in a monolithic way, as evidenced by the size of the LPs used for policy synthesis.

The runtimes include both the time required to create and the time to solve the optimization problems, and the AG runtimes account for the two optimization problems (one for each agent). Gurobi is used to solve the LPs; MONA is used to convert LTL$_f$ formulae into DFAs [24]. Results for both experiments are in Table I.

A. Experiment 1: Reach-Avoid Task on Gridworld

Policy synthesis for a reach-avoid task on the gridworld depicted in Fig. 1a is the first point of comparison. In this example, the two agents $\mathscr{M}^1$ and $\mathscr{M}^2$ initially start in the "northwest" corner of the gridworld at location (0, 0). The objective of $\mathscr{M}^1$ is to eventually reach its goal state marked by a while avoiding the obstacle at location b. Similarly, $\mathscr{M}^2$ attempts to reach c while avoiding d. These missions are formalized by the LTL$_f$ specifications:

$$\begin{aligned} \varphi^1 &: \quad \mathbf{F}(a) \land \mathbf{G}(\neg b) \\ \varphi^2 &: \quad \mathbf{F}(c) \land \mathbf{G}(\neg d). \end{aligned} \qquad (22)$$

The probability satisfaction thresholds are chosen as $(1-\delta^1) = (1-\delta^2) = 0.9$ for the individual agents, while a threshold of $(1-\delta) = 0.8$ is set for the conjoined specification. At every time step, agents have five available actions: four movement actions aligned to the cardinal directions (N, E, S, W) and a fifth `STAY` action. On taking the `STAY` action, the agent remains in the same location w.p. 1. For every movement action taken, let the probability of the agent actually *moving* in that direction be $p_\star$. The agent moves in each adjacent direction of the chosen action w.p. $(1-p_\star)/2$. It is impossible to move directly opposite to the chosen action. If the agent tries to move in a direction that is not possible (i.e., facing into edge of the gridworld), then the agent remains in its original location. Agent $\mathscr{M}^1$ has a $p_\star$ value of 0.9, while $\mathscr{M}^2$ is slightly less reliable with $p_\star = 0.8$.

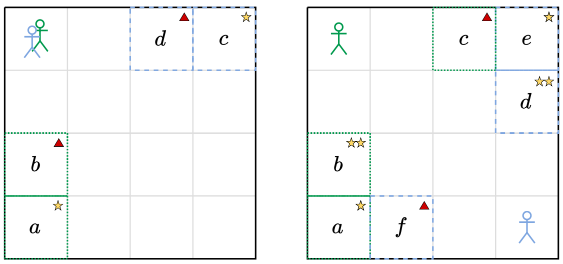
The rewards are defined jointly and encourage exploration by returning greater rewards for joint states in which $\mathscr{M}^1$ and $\mathscr{M}^2$ occupy different locations as shown

$$r_h(s^1, s^2, a^1, a^2) = \begin{cases} 2 & \text{if } s^1 \neq s^2 \\ 1 & \text{if } s^1 = s^2 \end{cases} \quad \forall\, h, a^1, a^2. \qquad (23)$$

The episode length (H) is 15 for the 4×4 sized gridworld, and it is incremented by one for every additional row added to the gridworld. For the larger gridworlds, the locations of the objectives and starting agent positions change to keep the same configuration with respect to the "edges" for the gridworld, i.e., a is always in the southwest corner with b located one space to the north. The same holds for c and the northeast corner. We assess the scalability of our solution by varying the size of the gridworld and comparing the policy synthesis time against the monolithic solution approach.

Gridworld Size	LP Size (vars, cons)		Runtime (s)			Achieved Reward		
	Monolithic	AG	Monolithic	AG	Speedup	Monolithic	AG	Relative Optimality
4x4	(512000, 20481)	(4609, 4609)	111.23	2.18	50.96 ×	30.95	30.29	97.86 %
5x5	(1328125, 53126)	(7651, 7651)	692.42	4.96	139.64 ×	32.95	32.29	97.99 %
6x6	(2916000, 116641)	(11665, 11665)	3094.31	11.47	269.68 ×	34.95	34.32	98.20 %
7x7	(5702375, 228096)	(16759, 16759)	10983.09	22.76	482.49 ×	36.95	36.37	98.42 %
8x8	(10240000, 409601)	(23041, 23041)	34181.37	57.47	594.75 ×	38.95	38.40	98.59 %
4x4	(1024000, 40961)	(6145, 6145)	426.96	3.59	119.08 ×	32.00	31.56	98.62 %
5x5	(2656250, 106251)	(10201, 10201)	2685.76	8.29	323.90 ×	34.00	33.84	99.54 %
6x6	(5832000, 233281)	(15553, 15553)	12393.80	18.06	686.18 ×	36.00	35.94	99.84 %

TABLE I: Comparison of AG approach to monolithic for Experiment 1 (top) and Experiment 2 (bottom)



(a) Experiment 1. (b) Experiment 2.

Fig. 1: Gridworlds used in the 4×4 experiments. Goal (reach) and obstacle (avoid) states corresponding to the LTL$_f$ specification are denoted by stars and triangles, respectively. Locations with fewer stars should be visited first. The starting positions of the agents are marked by the stick figures. Positions corresponding to $\mathscr{M}^1$ are marked in green, while those for $\mathscr{M}^2$ are shown in light blue.

B. Experiment 2: Ordered-Goal Navigation

The second gridworld experiment features the same transition dynamics and episode lengths as Experiment 1, and the notable locations (initial positions and labeled locations) shift in the same manner as the grid size increases. In this experiment, both agents have $p_\star$ set to 0.95. The probability satisfaction thresholds are set to $(1-\delta^1) = (1-\delta^2) = 0.95$ and $(1-\delta) = 0.9$. The mission specification for $\mathscr{M}^1$ is to first visit a, then proceed to location b while always avoiding c, as shown below:

$$\begin{aligned} \varphi^1 &: \quad \mathbf{F}(b) \wedge \mathbf{G}(\neg c) \wedge (\neg b \ \mathbf{U} \ a) \\ \varphi^2 &: \quad \mathbf{F}(e) \wedge \mathbf{G}(\neg f) \wedge (\neg e \ \mathbf{U} \ d). \end{aligned} \tag{24}$$

This simulates autonomous agents collecting and moving items to a new location (e.g., cargo to a warehouse) while navigating around various obstacles. The reward function is unchanged from the first experiment (23), again encouraging separation of the agents to avoid over-crowding.

All experimental results demonstrate a relatively small optimality gap between the AG-based decomposition solution and the monolithic approach. Furthermore, this optimality gap is shown to tighten as the problem size grows, while the speedup enjoyed by the AG approach continues to improve with respect to the monolithic problem.

VI. Conclusion

We have introduced a novel, assume-guarantee (AG) based methodology to split and solve logically constrained MDPs for multi-agent systems with significant scalability improvements and an empirically tight optimality gap. The AG decomposition blunts the conservatism while providing provable guarantees on logical constraint satisfaction. Analytical quantification of the optimality gap and the extension to unknown transition dynamics remain as future work.

References

[1] C.-H. Lo and N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *IEEE Trans. on Smart Grid*, Mar. 2013.

[2] M. L. Di Silvestre *et al.*, "How decarbonization, digitalization and decentralization are changing key power infrastructures," *Renewable and Sustain. Energy Reviews*, Oct. 2018.

[3] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. John Wiley & Sons, Inc., 1994.

[4] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999.

[5] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Int. Joint Conf. on Artif. Intell. (IJCAI)*, 2013.

[6] S. Zhu, L. M. Tabajara, J. Li, G. Pu, and M. Y. Vardi, "Symbolic LTLf synthesis," in *Int. Joint Conf. on Artif. Intell. (IJCAI)*, 2017.

[7] K. C. Kalagarla *et al.*, "Optimal control of logically constrained partially observable and multi-agent Markov decision processes," *IEEE Trans. on Autom. Control*, 2024.

[8] C. Daoui, M. Abbad, and M. Tkiouat, "Exact decomposition approaches for Markov decision processes: A survey," *Advances in Operations Res.*, 2010.

[9] J. Shen, X. Ye, and D. Feng, "A game-theoretic method for resilient control design in industrial multi-agent CPSs with Markovian and coupled dynamics," *Int. J. of Control*, Nov. 2021.

[10] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice-Hall, 1998.

[11] J. Castellini *et al.*, "Analysing factorizations of action-value networks for cooperative multi-agent reinforcement learning," *Auton. Agents and Multi-Agent Syst.*, Oct. 2021.

[12] L. Buşoniu, R. Babuška, and B. De Schutter, *Multi-agent Reinforcement Learning: An Overview*, ser. Stud. in Comput. Intell., 2010.

[13] A. Benveniste *et al.*, "Contracts for system design," *Foundations and Trends in Electron. Des. Automat.*, 2018.

[14] S. Bauer *et al.*, "Moving from specifications to contracts in component-based design," in *Fundam. Approaches to Softw. Eng.*, 2012.

[15] K. C. Kalagarla, R. Jain, and P. Nuzzo, "A Sample-Efficient Algorithm for Episodic Finite-Horizon MDP with Constraints," in *Proc. of the AAAI Conf. on Artif. Intell.*, 2021.

[16] A. Pnueli, "The temporal logic of programs," in *Annu. Symp. on Foundations of Comput. Sci. (sfcs 1977)*, 1977.

[17] P. Nuzzo *et al.*, "CHASE: Contract-based requirement engineering for cyber-physical system design," in *Des., Automat. & Test in Eur. Conf. & Exhib. (DATE)*, 2018.

[18] ——, "A contract-based methodology for aircraft electric power system design," *IEEE Access*, 2014.

[19] L. Bakule, "Decentralized control: An overview," *Annu. Reviews in Control*, Apr. 2008.

[20] D. D. Siljak, *Decentralized Control of Complex Systems*. Courier Corporation, Dec. 2011.

[21] I. Averbakh and V. Lebedev, "On the complexity of minmax regret linear programming," *Eur. J. of Oper. Res.*, Jan. 2005.

[22] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ. Press, 2004.

[23] J. Gallier, "Introduction to the theory of computation: Some notes for cis262," 2017. [Online]. Available: https://www.cis.upenn.edu/~cis2620/notes/tcbook-u.pdf

[24] N. Klarlund and A. Møller, *MONA Version 1.4 User Manual*, BRICS, Dept. of Comput. Sci., Univ. of Aarhus, January 2001. [Online]. Available: http://www.brics.dk/mona