Formal Synthesis of Control Policies for Continuous Time Markov Processes From Time-Bounded Temporal Logic Specifications

Ana Medina Ayala, Sean B. Andersson, and Calin Belta

***Abstract*—We consider the control synthesis problem for continuous-time Markov decision processes (CTMDPs), whose expected behaviors are measured by the satisfaction of Continuous Stochastic Logic (CSL) formulas. We present an extension of the CSL for CTMDPs involving sequential task specifications with continuous time constraints on the execution of those tasks. We then exploit model checking and time-bounded reachability algorithms to solve the CSL control synthesis problem. To illustrate the feasibility and effectiveness of our approach, we apply our methods to generate the optimal rate constants of a coupled set of biochemical reactions.**

***Index Terms*—Decision making, formal verification, Markov processes, stochastic systems.**

I. Introduction

Ongoing diversification of formal methods is paving the way for their use in areas that at least initially appeared to be unrelated to temporal logics and model checking. Given their expressivity and resemblance to natural language, temporal logics are a viable choice to enforce fundamental behaviors on controllable systems. On account of the existence of off-the-shelf tools, adapted model checking and automata game techniques can be used for verification [1]–[4] and control synthesis [5]–[7].

In this technical note, we focus on the control synthesis of continuous-time stochastic systems subject to time-sensitive requirements. In order to cope with specifications that depend crucially on timing, a quantitative notion of time must be incorporated both into the system's dynamics and also into the specification of the tasks to be performed by the system. Hence, we use continuous-time Markov decision processes (CTMDPs) to model the dynamics of such systems and specify the systems' requirements as Continuous Stochastic Logic (CSL) formulas. CTMDPs are stochastic models that may exhibit nondeterminism between transitions and in which the transition times are exponentially distributed. In CTMDPs, even though the state trajectories are functions of time, the system state can only take discrete values. CTMDPs have been used in a broad range of applications, from modeling controlled queuing systems in telecommunication and computer networks [8]–[10], to representing the dynamics of cellular processes in biology [11], [12], and to characterizing the stochastic behavior of autonomous agents in robotic applications [13], [14].

Unlike approaches that focus on model checking the behavior of CTMDPs from temporal logic specifications [15], in this work we present a complete framework for the explicit and automatic synthesis of control policies for CTMDPs from CSL formulas. Specifically, given a stochastic system modeled as a CTMDP, and a CSL formula over some properties satisfied at the states of the system, we obtain control policies that optimally satisfy the requirement expressed by the formula. Our approach to solve the control synthesis problem when only probabilistic operators are involved is based on an adaptation of the model checking techniques given in [16] and a recent algorithm [17] to compute approximate optimal control policies for continuous-time Markov games. If the long-run average operator is present in the formula, we exploit the graph structure of the CTMDP and reduce the control synthesis problem to solving a set of linear programming problems. Besides considering formulas based on state and path operators, our framework includes the control synthesis from nested formulas with concatenated timed intervals. In this case, our methods to solve the control synthesis problem are built from the algorithms derived for the probabilistic path and long-run average operators.

This work can be seen in the context of literature focused on the construction of optimal policies from temporal logic specifications and it is closely related to [7]. However, the focus in [7] is on generating optimal control policies from Probabilistic Computation Tree Logic (PCTL) formulas for discrete-time stochastic systems, here we focus on the problem of obtaining such policies for continuous-time systems. The analysis of CTMDPs for reachability properties has been previously considered in the literature [17]–[21]. In this technical note, we significantly expand the class of properties that can be handled by allowing for arbitrary CSL specifications and including nested and time-sequential formulas. Some preliminary results of this work, covering only timed specifications involving a single probabilistic operator, can be found in [22].

To illustrate the method developed in this technical note, we consider a stochastic model for a coupled set of biochemical reactions, which is formulated as a CTMDP in [23]. Assuming that the rate constants involved in these reactions can be controlled, we consider different time-constrained temporal logic specifications expressing requirements on the concentration levels of the molecules involved in the reaction.

II. Preliminaries

In this section, some concepts and notation used through the technical note are introduced.

Definition 1: A continuous-time Markov decision process (CTMDP) $\mathcal{M}$ is a tuple $(S, s_0, Act, \mathbf{R}, P, AP, L)$, where S is a finite set of states, $s_0 \in S$ is the initial state, Act is a finite nonempty set of actions, $\mathbf{R} : S \times Act \times S \to \mathbb{R}_{\geq 0}$ is a rate function such that for each $s \in S$ there is a pair $(\alpha, s') \in Act \times S$ with $\mathbf{R}(s, \alpha, s') > 0$, $P : S \times Act \times S \to [0, 1]$ is a transition probability function such that for each $s \in S$ and $\alpha \in Act$, either $P(s, \alpha, .)$ is a probability distribution on S or $P(s, \alpha, .)$ is the null function (i.e. $P(s, \alpha, s') = 0$ for any $s' \in S$), AP is a fixed set of atomic propositions, and $L : S \to 2^{AP}$ is a labeling function.

Given $s \in S$, with a slight abuse of notation, we use $Act(s)$ to denote the set of actions available at state s. We assume that each state has at least one outgoing transition. For a CTMDP $\mathcal{M}$, as defined above, the exit rate is defined as $E(s, \alpha) = \sum_{s' \in S} \mathbf{R}(s, \alpha, s')$.

A path ω of a CTMDP $\mathcal{M}$ is an infinite sequence $s_0 \xrightarrow{\alpha_0, t_0} s_1 \xrightarrow{\alpha_1, t_1} s_2 \xrightarrow{\alpha_2, t_2} \ldots$, where $s_i \in S$ is a state, $\alpha_i \in Act(s_i)$ is an action, $P(s_i, \alpha, s_{i+1}) > 0$, and $t_i \in \mathbb{R}_{\geq 0}$ is the sojourn time in state s_i. Any finite prefix of ω that ends in a state is a finite path of $\mathcal{M}$. The set of all non-empty finite sequences of paths is denoted by $Path^{fin}$ and that of infinite ones by $Path^{inf}$. For an infinite path ω and $i \in \mathbb{N}$, let $\omega[i] = s_i$ and $\delta(\omega, i)$ denote the $(i+1)$st state of ω and the time spent in s_i, respectively. For $t \in \mathbb{R}_{\geq 0}$, $\omega@t$ represents the state of the path ω that is occupied at time t.

Manuscript received March 21, 2012; revised February 1, 2013, October 20, 2013, and February 3, 2014; accepted February 25, 2014. Date of publication February 27, 2014; date of current version August 20, 2014. This work was supported in part by Boston University by the NSF under grants CNS-1035588 and CMMI-0928776, and the ONR MURI under Grant N00014-09-1051. Recommended by Associate Editor C. Szepesvári.

The authors are with the Department of Mechanical Engineering at Boston University, MA, 02215 USA (e-mail: duvinci@bu.edu; sanderss@bu.edu; cbelta@bu.edu).

Digital Object Identifier 10.1109/TAC.2014.2309033

In order to resolve the nondeterminism that occurs in the states of a CTMDP in which more than one action is allowed, a mapping from the paths to actions of the model is provided by a policy. In particular, to solve the control synthesis problem from specifications expressing the time bounded reachability probability, we restrict our attention to the following class of cylindrical policies.

Definition 2: [17] For a given timed path $\omega \in Path^{fin}$ and finitely many intervals that form a partition $\mathcal{J}$ of time, $[\omega]_{\mathcal{J}}$ contains all paths whose transition times are in the same equivalence class (with respect to $\mathcal{J}$) as those of ω. A policy is called cylindrical if, for some partition $\mathcal{J}$ of time into intervals, it has constant decisions for all paths ω, ω' having the same cylindrification ($[\omega]_{\mathcal{J}} = [\omega']_{\mathcal{J}}$).

Due to space limitations, we refer the readers to [17] for a detailed explanation on the construction of cylindrical policies.

Continuous Stochastic Logic (CSL) [16] is a branching time temporal logic based on Computation Tree Logic (CTL) [24]. A CSL formula sets conditions over the states of a CTMDP. For representing the syntax of CSL, we distinguish between state formulas (Φ) and path formulas (ϕ), which are evaluated over states and paths, respectively. Besides the standard propositional temporal logic operators, CSL includes the probabilistic operator $\mathbb{P}_{\sim p}(\phi)$ and the long-run average operator $\mathbb{S}_{\sim p}(\Phi)$. $\mathbb{P}_{\sim p}(\phi)$ asserts that the probability measure of the paths satisfying ϕ falls in the interval $I_{\sim p}$, where $I_{\sim p} = \{q \in [0,1] | q \sim p\}$ and $\sim \in \{<, \leq, >, \geq\}$. $\mathbb{S}_{\sim p}(\Phi)$ expresses that the long-run average fraction of time for a state satisfying Φ falls in the interval $I_{\sim p}$. The path formulas ϕ are defined as for CTL, except that a time-bounded next operator $\mathcal{X}^{\leq T}\Phi$ and a time-bounded until operator, $\Phi\, \mathcal{U}^{\leq T}\Psi$ for $T \geq 0$ are included. Intuitively, the path formula $\mathcal{X}^{\leq T}\Phi$ asserts that the next state transition occurs in at most T time units and that the next state must satisfy Φ. Similarly, $\Phi\, \mathcal{U}^{\leq T}\Psi$ expresses that a state satisfying Ψ must be reached in at most T time units via a path that satisfies Φ. Formally, we have:

Definition 3: The syntax of CSL state formulas is defined according to the following grammar rules:

$$\Phi ::= true | a | \neg\Phi | \Phi_1 \vee \Phi_2 | \Phi_1 \wedge \Phi_2 \, | \mathbb{P}_{\sim p}[\phi] | \, \mathbb{S}_{\sim q}[\Phi]$$

where $a \in AP$, ϕ is a path formula, $\sim \in \{<, \leq, >, \geq\}$ is a comparison operator, and $p, q \in [0,1]$ are thresholds. CSL path formulas are given by:

$$\phi ::= \mathcal{X}^{\leq T}\Phi | \Phi_1 \, \mathcal{U}^{\leq T} \Phi_2$$

where Φ, Φ_1, and Φ_2 are state formulas and $T \in \mathbb{R}_{\geq 0} \cup \{\infty\}$.

Definition 4: The validity of a CSL formula, relative to a CTMDP $\mathcal{M}$, is defined in terms of the satisfaction relation $\models$

$$\begin{aligned}
s &\models true && \forall s \in S \\
s &\models a && \text{iff } a \in L(s) \\
s &\models \neg\Phi && \text{iff } s \not\models \Phi \\
s &\models \Phi \wedge \Psi && \text{iff } s \models \Phi \wedge s \models \Psi \\
s &\models \mathbb{P}_{\sim p}[\phi] && \text{iff } p^s_\pi(\phi) \sim p \\
s &\models \mathbb{S}_{\sim q}[\Phi] && \text{iff } \tilde{q}\,(s, Sat(\Phi)) \sim q
\end{aligned}$$

where $p^s_\pi(\phi)$ denotes the probability measure of all the infinite paths that start at s and satisfy ϕ under policy π, and $\tilde{q}(s, Sat(\Phi))$ represents the long-run average fraction of time of states that satisfy the formula Φ, i.e., $Sat(\Phi) = \{s \in S | s \models \Phi\}$. Analogously, the satisfaction relation for CSL path formulas is defined by

$$\begin{aligned}
\omega &\models \mathcal{X}^{\leq T}\Phi && \text{iff } \exists \omega[1] \; s.t. \; \omega[1] \models \Phi \wedge \delta(\omega, 0) \leq T \\
\omega &\models \Phi \, \mathcal{U}^{\leq T} \Psi && \text{iff } \exists T' \leq T \; s.t. \; \omega@T' \models \Psi \\
& && (\forall T'' < T' \; s.t. \; \omega@T'' \models \Psi).
\end{aligned}$$

The other standard boolean connectives can be derived from the semantics given above as $false = \neg true$, $\Phi \vee \Psi = \neg(\neg\Phi \wedge \neg\Psi)$, and $\Phi \rightarrow \Psi = \neg\Phi \vee \Psi$. The time-bounded versions of the eventually ($\Diamond$) and always ($\Box$) operators can be formulated as $\mathbb{P}_{\sim p}(\Diamond^{\leq T}\Phi) = \mathbb{P}_{\sim p}(true \;\; \mathcal{U}^{\leq T}\Phi)$ and $\mathbb{P}_{\geq p}(\Box^{\leq T}\Phi) = \mathbb{P}_{\leq 1-p}(\Diamond^{\leq T}\neg\Phi)$, respectively.

The maximal (minimal) probabilistic operator $\mathbb{P}_{\max=?}\phi$ ($\mathbb{P}_{\min=?}\phi$) is defined as the maximal (minimal) probability with which ϕ can be satisfied by a policy. More formally we have:

$$\mathbb{P}_{\max=?} = \sup_{\pi \in \Pi} p^s_\pi(\phi), \quad \mathbb{P}_{\min=?} = \inf_{\pi \in \Pi} p^s_\pi(\phi).$$

The defined measures are also referred to as optimal probabilities. A policy π is optimal if through its execution the optimal probability is achieved. The optimal average fraction of time and corresponding policies are defined for the long run average operator in a similar way. In this case, we define

$$\mathbb{S}_{\max=?} = \sup_{\pi \in \Pi} \tilde{q}\,(s, Sat(\Phi)), \quad \mathbb{S}_{\min=?} = \inf_{\pi \in \Pi} \tilde{q}\,(s, Sat(\Phi)).$$

The defined CSL probabilistic path and long-run average operators constitute the basis to express more complex behaviors for a given system. To provide more expressivity, we can then combine probabilistic and long-run average formulas into *Nested* CSL formulas. An example of such formulas is given in (12).

III. Problem Formulation and Approach

The problem that we consider in this technical note can be formulated as follows:

Problem 1: Given a continuous-time Markov decision process model $\mathcal{M}$ (Def. 1) and a CSL specification path formula ϕ or state formula Φ (Def. 3), find a control strategy that:

- maximizes the probability of satisfying ϕ, or
- maximizes the long-run average fraction of time of satisfying Φ. □

Briefly, our approach to solve *Problem 1* generates the control policy for which a CSL formula is optimally satisfied. In a manner standard for model checking, we construct the parse tree of the CSL path formula ϕ (state formula Φ), recursively computing the set of states that satisfies ϕ (Φ). Each one of the probabilistic path and long-run average operators in ϕ (Φ) is considered individually.

Distinct control synthesis algorithms are derived for each probabilistic path operator and for the long run average formula. These algorithms are then used as the building blocks to find a control strategy for formulas that contain more than one path formula or a combination of path and long-run average formulas. In order to cope with time in specifications that involve temporal sequencing in the latter class of formulas, we introduce the notion of a *reset* event. A reset event indicates the completion of a task delimited by a certain period of time. We consider time as a global variable that is reset with the occurrence of a reset event. The control algorithms for probabilistic path, long-run average, and nested formulas are detailed in the following section.

IV. Control Synthesis of CTMDPs From CSL Specifications

In order to solve the control synthesis problem of probabilistic path formulas, we take advantage of existing algorithms used in the formal methods community and adapt them to find a policy that produces the maximal probability of satisfying a formula. Our approach to solve long-run average formulas is given as the solution of a set of optimization problems created by considering the CTMDP model as a graph. The solutions to the control synthesis problems for the probabilistic path and long-run average formulas will be the basis to solve nested formulas.

A. Probabilistic Path Formulas

1) Time-Bounded Next Optimal Operator ($\mathbb{P}_{\max=?}[\mathcal{X}^{\leq T}\Phi]$)*:* For these type of formulas, we seek a policy that produces the maximal probability of satisfying the given formula as well as the value of this probability. This is achieved by modifying an existing model checking algorithm [16]. Let s denote the current state of the system. The solution to finding the optimal action to satisfy the formula in at most T time units with exactly one transition can be reduced to:

$$\pi^*(s) = \arg\max_{\alpha\in Act(s)} \left(1 - e^{-\mathbf{E}(s,\alpha)\cdot T}\right) \sum_{s'\models\Phi} P(s,\alpha,s'). \quad (1)$$

The obtained optimal policy is stationary and can be computed using a dynamic programming algorithm. The solution for $\mathbb{P}_{\min=?}[\mathcal{X}^{\leq T}\Phi]$ is found in a similar manner by defining a minimization problem.

2) Time-Bounded Until Optimal Operator ($\mathbb{P}_{max=?}[\Phi\ \mathcal{U}^{\leq T}\Psi]$)*:* The maximum probability of satisfying $\Phi\,\mathcal{U}^{\leq T}\Psi$ is achieved by means of an adaptation of the approach described in [17] to approximate optimal control within a Markov game. The advantage of this method is that it can be applied to general CTMDPs, eliminating the need for locally uniform models [20]. Let $p_{\max}(s,x)$ be the maximum probability of satisfying the time-bounded until formula, within T time units, starting from state s given that x time units have passed. It follows that $p_{\max}(s,x)$ can be characterized by the following set of Bellman equations:

$$p_{\max}(s,x) = \begin{cases} \max\limits_{\alpha\in Act(s)} \left\{ \int_0^{T-x} \sum\limits_{s'\in S} \mathbf{E}(s,\alpha)\cdot e^{-\mathbf{E}(s,\alpha)\cdot x} \right. & \\ \qquad \left. \cdot P(s,\alpha,s')\cdot p_{\max}(s',x+\tau)\mathrm{d}\tau \right\} & \text{if } s\models\Phi\wedge\neg\Psi \\ & \text{and } x\leq T \\ 1 & \text{if } s\models\Psi \\ 0 & \text{otherwise.} \end{cases}$$

One way to solve this Bellman equation is to reduce it to a system of ordinary differential equations with decisions. To this end, let $p(s,t)$ be the maximum probability to reach a state that satisfies Ψ from state s given that t time units have passed already. In the sequel, let $\mathbf{Q}(s,\alpha,s') = \mathbf{E}(s,\alpha)\cdot P(s,\alpha,s')$ if $s'\neq s$ and $\mathbf{Q}(s,\alpha,s) = -\sum_{s'\neq s}\mathbf{E}(s,\alpha)\cdot P(s,\alpha,s') + \mathbf{E}(s,\alpha)\cdot P(s,\alpha,s)$. For any $t<T$, we define

$$-\dot{p}(s,t) = \max_{\alpha\in Act(s)} \left\{ \sum_{s'\in S} \mathbf{Q}(s,\alpha,s')\cdot(p(s',t)-p(s,t)) \right\}. \quad (2)$$

The key of the approach presented in [17] is to break time into intervals of length ϵ, and to independently solve for (approximate) optimal controls in each of the resulting T/ϵ intervals. As a result, by means of a piecewise polynomial function p_1 as an approximation, the optimal action at the start of each interval is obtained. Such an action is assumed to be optimal for the duration of the interval.

We start by defining $p_1(s,T)=1$ if $s\models\Psi$ and $p_1(s,T)=0$, otherwise. If p_1 is defined for the interval $[t,T]$, the following procedure is used to extend it to the interval $[t-\epsilon,T]$. First, for all $s\in S$, an action that maximizes the right-hand side of the previous equation is chosen. Then, the descent of $p_1(s,\cdot)$ within the interval $[t-\epsilon,t]$ is fixed as $c(s,t)=\sum_{s'\in S}\mathbf{Q}(s,\alpha,s')\cdot p_1(s',t)$. For every $\tau\in[0,\epsilon]$, and every $s\in S$, we have:

$$-\dot{p}_1(s,t-\tau) = c(s,t)$$

yielding

$$p_1(s,t-\tau) = p_1(s,t) + \tau\cdot c(s,t).$$

Thus, the optimal policy is chosen by solving for p_1 during the interval $[t-\epsilon,t]$. For each $s\in S$, and $\tau\in[t-\epsilon,t]$, we can define a system of differential equations $g_1(s,\tau)$ as

$$-\dot{g}_1(s,\tau) = \sum_{s'\in S}\mathbf{Q}(s,\alpha,s')\cdot g_1(s',\tau). \quad (3)$$

The system of differential equations defined above gives the maximum probability obtained by following the memoryless policy that is derived from p_1. The solution for $\mathbb{P}_{\min=?}[\Phi\mathcal{U}^{\leq T}\Psi]$ is found in an analogous manner.

Remark 1: The presented approach, introduced in [17] as the single ϵ-net technique, can be seen as the starting point to approximate optimal policies while maintaining high precision. Similarly, one can use higher order polynomial approximations to derivate p_1 within each interval $[t-\epsilon,t]$ by means of double, triple or quadruple ϵ-nets.

B. Long-Run Average Formulas

Our solution to the synthesis problem of formulas of the form $\mathbb{S}_{\max=?}[\Phi]$ relies on the graphical structure of the CTMPD $\mathcal{M}$. Maximal end-components (MEC), introduced in [25], play an important role in solving the synthesis problem of CSL formulas involving the long-run average operator. Given a CTMDP $\mathcal{M}=(S,s_0,Act,\mathbf{R},P,AP,L)$, let $MEC(\mathcal{M})$ denote the set of MECs of $\mathcal{M}$. Let $\tau_t^{\mathcal{M}}$ be the average fraction of time spent at any state satisfying Φ up to time t. For a given policy π, we define the long-run average fraction of time that Φ is satisfied, considering $t\to\infty$, as

$$\lim_{N\to\infty}\mathbb{E}\left[\tau_{t_N}^{\mathcal{M}}\right] = \sum_{C\in MEC(\mathcal{M})}\left(p_\pi^{s_0}(\Diamond C)\cdot\lim_{N\to\infty}\mathbb{E}\left[\tau_{t_N}^{C}\right]\right). \quad (4)$$

Our approach starts by recursively determining $Sat(\Phi)=\{s\in S|s\models\Phi\}$, i.e., the set of states that satisfy Φ. Subsequently, the set of MECs, $MEC(\mathcal{M})$, is obtained using a graph algorithm for computing end-components [26]. Our method to compute the expected value of the long-run average fraction of time consists of two steps. In the first step, the long-run average throughput is modeled by the average reward to be accumulated in the long-term. Thus, if $s\models\Phi$, such a reward is defined as the expected single-stage reward given that the system occupies state s and the decision maker chooses action α, corresponding to the (s,α) pair. Then, the average reward is given by the expected value of the total residence time distribution in s under action α, that is

$$r(s,\alpha) = \int_0^\infty E(s,\alpha)t\cdot e^{-E(s,\alpha)t}\sum_{s'\in S}P(s,\alpha,s')\mathrm{d}t$$

otherwise, it is equal to 0.

Let $J^*(s)$ denote the optimal average reward accumulated in the long-run starting at s and executing the optimal policy. Then, for all $s \in S$, it can be shown [27] that there exists a scalar λ^* such that $J^*(s) = \lambda^*$. Furthermore, there is a function $h^*(s)$, for all $s \in S$ such that

$$h^*(s) = \max_{\alpha \in Act(s)} \left[r(s,\alpha) - \frac{\lambda^*}{E(s,\alpha)} + \sum_{s' \in S} P(s,\alpha,s') \right] \quad (5)$$

which can be translated to the following linear programming (LP) formulation:

Maximize λ subject to

$$h(s) + \frac{\lambda}{E(s,\alpha)} \leq r(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s')h(s') \quad (6)$$

with $\lambda \cup \{h(s) | s \in S\}$ as variables.

Let B represent the set of states that belong to any $MEC(\mathcal{M})$. In the second step of our approach, we formulate the following LP problem to account for the probability of eventually reaching a MEC in (4), denoted by $p(s)$:

Minimize $\sum_{s' \in S \setminus B} p(s)$ subject to:

$$p(s) \geq \sum_{s' \in B} P(s,\alpha,s')\lambda_B + \sum_{s' \in S \setminus B} P(s,\alpha,s')p(s'). \quad (7)$$

The solution to this LP problem corresponds to the probabilities that maximize the average long-run reward of eventually reaching some state in B from s_0. The actions that generate these probabilities together with the actions that belong to the MECs constitute the optimal policy. For the long-run average operator, the optimal policy is stationary.

Similarly, for $\mathbb{S}_{\min=?}[\Phi]$, we can determine the actions that generate the minimum long-run average fraction of time that is spent in some state in any $MEC(\mathcal{M})$ only if it comprises a state satisfying Φ. The solution to this problem is found using the same approach as for the $\mathbb{S}_{\max=?}$ case by first defining a minimization LP problem to find the optimal value of λ. The value of λ obtained from the solution of this LP problem is used to formulate a maximization LP problem from which the optimal policy can be obtained.

C. Nested Formulas

By combining probabilistic and long-run average operators, the expressivity of CSL formulas can be enhanced. The required condition to express complex properties by means of nested probabilistic and long-run average formulas is that all inner $\mathbb{P}$ and $\mathbb{S}$ operators are of the form $\mathbb{P}_{\sim p}[\phi]$ and $\mathbb{S}_{\sim q}[\Phi]$. This requirement is imposed due to the fact that for each nested operator, the set of states satisfying either ϕ or Φ needs to be determined. In general, there are three cases that we need to consider

$$\mathbb{P}_{max=?}[\mathcal{X}^{\leq T}\phi_A], \quad (8)$$

$$\mathbb{P}_{max=?}[\phi_B \, \mathcal{U}^{\leq T}\phi_A], \quad (9)$$

$$\mathbb{S}_{max=?}[\Phi_C] \quad (10)$$

where ϕ_B and ϕ_A may include either the $\mathbb{P}$ or the $\mathbb{S}$ operator, while Φ_C should be a $\mathbb{P}$ operator.

To compute a control strategy for a CTMDP $\mathcal{M}$ to satisfy a nested formula, we start by considering the case in which the outer-most operator is probabilistic, as in (8) or (9). Our approach is similar to the discrete setting developed in [7]. We start by determining the set of states satisfying ϕ_A. Subsequently, the corresponding control policy π_A is obtained by employing the algorithms given in Sections IV-A and B. Then, for formulas defined as in (9), the states that satisfy ϕ_B are identified and all the satisfying control policies π_B are acquired using the same methods as for finding π_A. However, it is important to establish that we consider all satisfying policies only for the time-bounded next operator. This restriction is due to the computational complexity of finding all the satisfying policies for the long-run average and time-bounded until operators. For these operators, we only consider the optimal control policy that generates the extremum probability of satisfying π_B.

A new CTMDP is then obtained by removing the non-satisfying policies. As a result of this procedure, the new CTMDP may include states for which no enabled transitions are available. If this is the case, these states are made absorbing. Finally, one of the algorithms that generates an optimal policy (Sections IV-A and B) is applied to the outer-most probabilistic operator on the new CTMDP. Consequently, the optimal control policy π, and the probability corresponding to this policy are obtained. Policy π is executed until a state that satisfies ϕ_A is reached within T or less time units, resetting the global clock. Afterwards, the control policy π_A is undertaken.

The case in which the long-run average is the outer-most operator is treated as follows. Let Φ_C in (10) be equal to $\mathbb{P}_{\sim p}[\phi]$, where ϕ is a path formula. By the semantics of CSL formulas, checking ϕ is equivalent to verifying whether the probability generated while applying a certain policy meets the bound $\sim p$, that is whether $p^s_\pi \sim p$ (Def. 4). Therefore, we start by determining $Sat(\phi)$, i.e., the set of states satisfying ϕ. Subsequently, the control policy π_C that satisfies the probability bound at each state is generated using the algorithms given in Section IV-A. The states that satisfy this bound under the attained policy constitute the set of states that satisfy ϕ. The label of a state s is extended by a_{Φ_C} if $s \in Sat(\Phi_C)$. In order to find the optimal policy for this nested formula, the approach of Section IV-B is used. A maximization LP program is formulated for each MEC that contains a state labeled with the atomic proposition a_Φ to obtain a scalar that satisfies (5). The policy that maximizes the fraction of time to be spent in a a_Φ-state is then generated from the solution of the LP problem in (7). This policy is applied until an a_{Φ_C}-labeled state is attained. At this point, the control policy π_C takes effect.

Remark 2: For specifications containing more than one time-bounded until operator, the interval length ϵ is defined *a priori* and remains constant for the overall specification, independently of the number of intervals defined for each until operator present in the formula.

D. Time Complexity

In this section, we discuss the time complexity of the formal synthesis of CSL formulas as well as some implementation considerations. Let $\mathcal{M} = (S, s_0, Act, \mathbf{R}, P, AP, L)$ be a CTMDP, $N = |S|$ be the number of states in $\mathcal{M}$ and M the number of nonzero entries in the transition probability matrix P. Since in the control synthesis of CSL formulas we consider each probabilistic path and long-run average operators independently, the time complexity is linear in the size of the formula.

The control synthesis of probabilistic formulas of the form $\mathcal{X}^{\leq T}\Phi$ requires $\mathcal{O}(M+N)$ time as $\mathcal{O}(M+N)$ multiplications are needed in the optimization procedure. For probabilistic formulas of the form $\Phi_1 \, \mathcal{U}^{\leq T}\Phi_2$, the time complexity of the algorithm is $\mathcal{O}(M \cdot (T/\epsilon) + N)$. For formulas involving the long-run average operator, a graph analysis is carried out to determine the MECs of $\mathcal{M}$. This takes $\mathcal{O}(M+N)$ time. In the worst case, for each identified MEC C, $|C|$ linear programming problems need to be solved once. Finally, the probability of reaching a MEC C needs to be computed for each MEC. This requires solving a linear programming problem, taking in the worst case $\mathcal{O}(N^3)$ time.

V. Case Study: Biochemical Reaction

Continuous-time Markov models have been extensively used for modeling biochemical reaction networks [28], [29]. In this section, we focus on the stochastic model for a generic biochemical system consisting of two species proposed in [23]. Let X_1 and X_2 be the number of molecules per cell of species 1 and 2, respectively. The corresponding biochemical reactions can be described by the following birth-death Markov process:

$$X_1 \xrightarrow{a} X_1 + 1 \quad X_2 \xrightarrow{cX_1} X_2 + 1$$

$$X_1 \xrightarrow{bX_1} X_1 - 1 \quad X_2 \xrightarrow{dX_2} X_2 - 1$$

where a, b, c, and d represent rate constants.

The CTMDP that models the described biochemical reactions is given in [23]. According to this model, a state at time $t \geq 0$ represents the number of molecules of each species, that is $s = (X_1, X_2)$. Therefore, the state space has the form of $S = \{1, 2, \ldots\}^2$.

The transition rates of the reactions can be expressed as

$$\mathbf{R}(s, \alpha, s') = \begin{cases} a & \text{if } s' = (X_1 + 1, X_2) \\ cX_1 & \text{if } s' = (X_1, X_2 + 1) \\ -(a + bX_1 + cX_1 + dX_2) & \text{if } s' = (X_1, X_2) \\ bX_1 & \text{if } s' = (X_1 - 1, X_2) \\ dX_2 & \text{if } s' = (X_1, X_2 - 1) \\ 0 & \text{otherwise.} \end{cases}$$

Assuming that the rate constants a, b, c, d can be controlled by a decision maker, $\alpha = (a, b, c, d)$ can be seen as an action on $Act(s)$ at state $s \in S$.

When analyzing biological systems at the lowest level of abstraction, one is concerned with the behavior of individual molecules rather than a bulk response. Following the setup in [23], we set the copy number of each of the molecular species of the model defined above to be five. Thus, the number of states representing the biochemical system equals 25. The set of available actions is defined as in [23]. We apply labels $AP = \{1, \ldots, n_k\}$, and for $0 \geq p \leq n_k$ let $L(s, p) = true$ if $s = (X_1, X_2)$ with $X_1 = p$ and $L(s, p) = false$, otherwise.

We consider two specifications for the defined biochemical reaction.

Specification 1: "In the long-run, the number of molecules per cell of species 1 is at least 4".

Specification 2: "Within 20 time units, the number of molecules per cell of species 1 reaches at least 3 and then at least 50% of the time with probability greater than or equal to 0.7, it reaches at least 5 in less than 10 time units".

In this technical note we examine the generation of policies that produce the maximum long-run average fraction of time to be spent in states satisfying specification 1 and the maximum probability of satisfying specification 2. These specifications translate to the CSL formulas

$$\phi_1 \quad :: \quad \mathbb{S}_{\max=?}[X_1 \geq 4], \tag{11}$$

$$\phi_2 \quad :: \quad \mathbb{P}_{\max=?}\left[true\, \mathcal{U}^{\leq 20}\, (X_1 \geq 3 \wedge \mathbb{S}_{\geq 0.5}\left[\mathbb{P}_{\leq 0.7}[true\, \mathcal{U}^{\leq 10} X_1 = 5]\right]\right)\right]. \tag{12}$$

The model described above as well as the solution to *Problem 1* developed in Section IV have been implemented in MATLAB, on a computer with 2-GHz dual processor. As a result, the average fraction of time of satisfying 1 was found to be 0.838, while the probability of satisfying 2 was 0.657. The running times to compute the optimal rates for 1 and 2 were 14.56 and 265.7 s, respectively.

Tables I and II show the resulting optimal rates a, b, c, and d for specifications 1 and 2, respectively. Note that given the inclusion of nested until operators in specification 2, the optimal rates generated for this specification are time dependent. The interval length ϵ used in this case was equal to 0.1. Due to space constraints, Table II only shows the control policy corresponding to the last interval. However, the optimal rates obtained at each interval for specification 2 did not differ significantly. As an example, Fig. 1 depicts the optimal rates a and c within the time interval given in specification 2 for the state represented by $X_1 = 1$, $X_2 = 1$. Rates b and d were constant and equal to 0 within the time interval.

TABLE I
Optimal Rates a, b, c, and d Per State (X_1, X_2) Generated for Specification 1

	$X_2 = 1$	$X_2 = 2$	$X_2 = 3$
$X_1 = 1$	[0.6 0.00 0.55 0]	[0.6 0.00 0.60 0.25]	[0.6 0.00 0.50 0.25]
$X_1 = 2$	[0.6 0.25 0.60 0]	[0.6 0.25 0.60 0.25]	[0.6 0.25 0.60 0.25]
$X_1 = 3$	[0.6 0.25 0.60 0]	[0.6 0.25 0.60 0.30]	[0.6 0.25 0.55 0.25]
$X_1 = 4$	[0.6 0.25 0.60 0]	[0.6 0.25 0.60 0.30]	[0.6 0.25 0.60 0.25]
$X_1 = 5$	[0.0 0.25 0.55 0]	[0.0 0.25 0.55 0.25]	[0.0 0.25 0.55 0.25]

	$X_2 = 4$	$X_2 = 5$
$X_1 = 1$	[0.6 0.00 0.50 0.25]	[0.6 0.00 0 0.25]
$X_1 = 2$	[0.6 0.25 0.60 0.25]	[0.6 0.25 0 0.25]
$X_1 = 3$	[0.6 0.25 0.60 0.25]	[0.6 0.25 0 0.30]
$X_1 = 4$	[0.6 0.30 0.55 0.25]	[0.6 0.30 0 0.25]
$X_1 = 5$	[0.0 0.25 0.00 0.25]	[0.0 0.25 0 0.25]

TABLE II
Optimal Rates a, b, c, and d Per State (X_1, X_2) Generated for Specification 2 Within the Last Time Interval

	$X_2 = 1$	$X_2 = 2$	$X_2 = 3$
$X_1 = 1$	[0.6 0.00 0.60 0]	[0.6 0.00 0.50 0.30]	[0.6 0.00 0.60 0.25]
$X_1 = 2$	[0.6 0.30 0.55 0]	[0.6 0.30 0.60 0.25]	[0.6 0.25 0.50 0.25]
$X_1 = 3$	[0.6 0.30 0.50 0]	[0.6 0.25 0.55 0.25]	[0.6 0.30 0.60 0.30]
$X_1 = 4$	[0.6 0.25 0.50 0]	[0.6 0.25 0.60 0.30]	[0.6 0.30 0.60 0.30]
$X_1 = 5$	[0.0 0.25 0.60 0]	[0.0 0.25 0.55 0.35]	[0.0 0.30 0.50 0.25]

	$X_2 = 4$	$X_2 = 5$
$X_1 = 1$	[0.6 0.00 0.50 0.30]	[0.6 0.00 0 0.30]
$X_1 = 2$	[0.6 0.25 0.55 0.30]	[0.6 0.30 0 0.25]
$X_1 = 3$	[0.6 0.25 0.55 0.30]	[0.6 0.25 0 0.25]
$X_1 = 4$	[0.6 0.25 0.55 0.25]	[0.6 0.25 0 0.30]
$X_1 = 5$	[0.0 0.25 0.00 0.25]	[0.0 0.30 0 0.25]

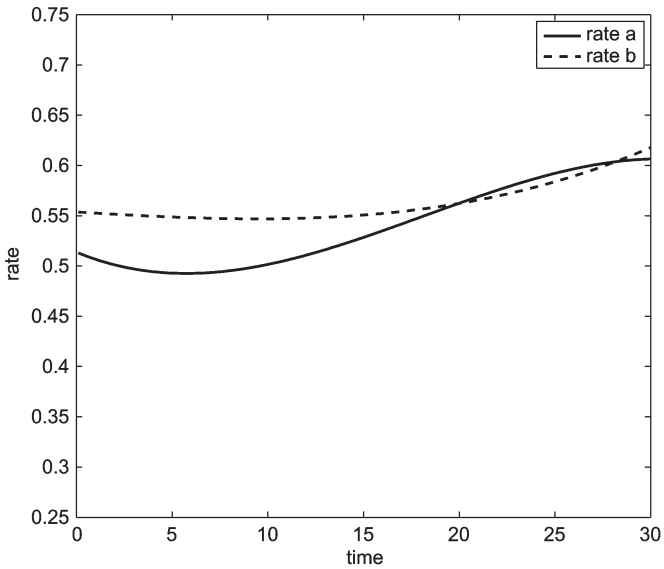

Fig. 1. Optimal rates a and c for state $X_1 = 1$, $X_2 = 1$ computed on the interval [0 30]. The values of these rates are captured by a solid line for a and a dotted line for c. Rates b and d are always equal to 0 for this particular state.

VI. Conclusions

This technical note presented a complete computational framework to solve the control synthesis for CSL formulas over CTMDPs. Algorithms were derived for the probabilistic path operator and the long-run average formula. They were then used as the basis to generate a control strategy for nested formulas. We illustrated our methods in the design of optimal rates for a set of biochemical reactions.

References

[1] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multiagent motion tasks based on LTL specifications," in *Proc. IEEE Conf. Decision and Control*, Dec. 2004, pp. 153–158.

[2] M. M. Quottrup, T. Bak, and R. Izadi-Zamanabadi, "Multi-robot motion planning: A timed automata approach," in *Proc. IEEE Int. Conf. Robot. and Autom.*, Apr. 2004, pp. 4417–4422.

[3] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Hybrid controllers for path planning: A temporal logic approach," in *Proc. IEEE Conf. Decision and Control*, Dec. 2005, pp. 4885–4890.

[4] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.

[5] H. Kress-Gazit, T. Wongpiromsarn, and U. Topcu, "Correct, reactive, high-level control: Mitigating the state explosion problem of temporal logic synthesis," *IEEE Robot. Automat. Mag.*, vol. 18, no. 3, pp. 65–74, Sep. 2011.

[6] X. C. Ding, M. Kloetzer, Y. Chen, and C. Belta, "Automatic deployment of robotic teams: An automata theoretic approach," *IEEE Robot. Automat. Mag.*, vol. 18, no. 3, pp. 75–86, Sept. 2011.

[7] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *IEEE Trans. Robotics*, vol. 28, no. 2, pp. 396–409, Apr. 2012.

[8] M. Y. Kitaev and V. V. Rykov, *Controlled Queueing Systems*. Boca Raton, FL: CRC Press., 1995.

[9] L. I. Sennout, *Stochastic Dynamic Programming and the Control of Queueing Systems*. New York: Wiley, 1999.

[10] L. Tadj and G. Choudhury, "Optimal design and control of queues," *TOP*, vol. 13, no. 2, pp. 359–412, Dec. 2005.

[11] M. Iosifescu and P. Tautu, *Stochastic Processes and Applications in Biology and Medicine: Theory*, vol. 2. Berlin, Germany: Springer, 1973.

[12] D. J. Wilkinson, "Stochastic modelling for quantitative description of heterogeneous biological systems," *Nature Rev. Genetics*, vol. 10, pp. 122–133, Feb. 2009.

[13] N. Napp and E. Klavins, "A compositional framework for programming stochastically interacting robots," *Int. J. Robot. Res.*, pp. 713–729, May 2011.

[14] M. Anderson, R. Fierro, A. Naseri, W. Lu, and S. Ferrari, "Cooperative navigation for heterogeneous autonomous vehicles via approximate dynamic programming," in *Proc. IEEE Conf. Decision and Control*, Dec. 2011.

[15] P. Buchholz, E. M. Hahn, H. Hermanns, and L. Zhang, "Model checking algorithms for CTMDPs," in *Proc. Int. Conf. Comput. Aided Verification*, July 2011, vol. 6806, pp. 225–242.

[16] C. Baier, B. Haverkort, H. Hermanns, and J. P. Katoen, "Model checking algorithms for continuous time Markov chains," *IEEE Trans. Software Eng.*, vol. 29, no. 7, pp. 524–541, June 2003.

[17] M. N. Rabe and S. Schewe, "Finite optimal control for time-bounded reachability in CTMDPs and continuous-time markov games," *Acta Inf.*, vol. 48, no. 5–6, pp. 291–315, Aug. 2011.

[18] C. Baier, H. Hermanns, J.-P. Katoen, and B. R. Haverkort, "Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes," *Theor. Comput. Sci.*, vol. 345, no. 1, pp. 2–26, Nov. 2005.

[19] T. Chen, T. Han, J. P. Katoen, and A. Mereacre, "Observing continuous-time MDPs by 1-clock timed automata," in *Proc. Workshop Reachability Problems*, Sept. 2011, pp. 2–25.

[20] M. R. Neuhäußer and L. Zhang, "Time-bounded reachability probabilities in continuous-time Markov decision processes," in *Proc. Int. Conf. Quantitative Evaluation of Syst.*, Sept. 2010, pp. 209–218.

[21] P. Buchholz and I. Schulz, "Numerical analysis of continuous time Markov decision processes over finite horizons," *Computers & OR*, vol. 38, no. 3, pp. 651–659, Mar. 2011.

[22] A. I. Medina Ayala, S. B. Andersson, and C. Belta, "Probabilistic control from time-bounded temporal logic specifications in dynamic environments," in *Proc. IEEE Int. Conf. Robot. and Autom.*, May 2012, pp. 4705–4710.

[23] X. Guo, Q. Lui, and T. Zhou, "Optimal control of stochastic fluctuations in biochemical reactions," *J. Biological Syst.*, vol. 17, no. 2, pp. 283–301, June 2009.

[24] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA: The MIT Press, 2008.

[25] L. de Alfaro, "How to specify and verify the long-run average behavior of probabilistic systems," in *Proc. Int. Symp. Logic in Comput. Sci.*, June 1998, pp. 454–465.

[26] M. Ummels, *Stochastic Multiplayer Games: Theory and Algorithms*. Amsterdam, The Netherlands: Pallas Publications, Amsterdam University Press, 2010.

[27] D. P. Bertsekas, *Dynamic Programming and Optimal Control, vol. 2*. Belmont, MA: Athena Scientific, 1995.

[28] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Mach. Learn.*, vol. 52, no. 1–2, pp. 169–191, Jan. 2004.

[29] P. C. Y. Chen and J. W. Chen, "A Markovian approach to the control of genetic regulatory networks," *Biosystems*, vol. 90, no. 2, pp. 535–545, Sept. 2007.