Systems & Control Letters 111 (2018) 40–48

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Safe Metropolis–Hastings algorithm and its application to swarm control ☆

Mahmoud El Chamie [a,*], Behçet Açıkmeşe [b]

[a] United Technologies Research Center (UTRC), Systems Department, East Hartford, CT 06108, USA
[b] University of Washington, William E. Boeing Department of Aeronautics and Astronautics, Seattle, WA 98195, USA

ARTICLE INFO

Article history:
Received 20 November 2016
Received in revised form 13 September 2017
Accepted 16 October 2017

Keywords:
Markov chains
Metropolis–Hastings
Safety constraints
Convex optimization
Swarm control

ABSTRACT

This paper presents a new method to synthesize *safe* reversible Markov chains via extending the classical Metropolis–Hastings (M–H) algorithm. The classical M–H algorithm does not impose *safety* upper bound constraints on the probability vector, discrete probability density function, that evolves with the resulting Markov chain. This paper presents a new M–H algorithm for Markov chain synthesis that ensures such safety constraints together with reversibility and convergence to a desired stationary (steady-state) distribution. Specifically, we provide a convex synthesis method that incorporates the safety constraints via designing the *proposal* matrix for the M–H algorithm. It is shown that the M–H algorithm with this proposal matrix, *safe* M–H algorithm, ensures safety for a well-characterized convex set of stationary probability distributions, i.e., it is robustly safe with respect to this set of stationary distributions. The size of the safe set is then incorporated in the design problem to further enhance the robustness of the synthesized M–H proposal matrix. Numerical simulations are provided to demonstrate that multi-agent systems, swarms, can utilize the safe M–H algorithm to control the swarm density distribution. The controlled swarm density tracks time-varying desired distributions, while satisfying the safety constraints. Numerical simulations suggest that there is insignificant trade-off between the speed of convergence and the robustness.

1. Introduction

The Metropolis–Hastings (M–H) algorithm [1–4] is a method for obtaining random samples from a probability distribution. The M–H algorithm builds on the theory of Markov processes and Markov-Chain-Monte-Carlo (MCMC) sampling methods [5–8] to synthesize reversible Markov chains that guarantee the desired stationary distributions. Recent research has focused on synthesizing fast mixing Markov chains with desired stationary distributions that incorporate constraints on transition probabilities [9,10] by using tools from graph theory [11], Lyapunov stability analysis [12], and convex optimization [13].

The M–H algorithm is very useful when online Markov chain synthesis is needed because it can be implemented easily and executed very efficiently. Given a matrix K (called proposal matrix), the M–H algorithm can be used to construct a stochastic transition matrix M of a Markov chain to satisfy some specifications, e.g., a prescribed stationary distribution and constraints on transitions. The matrix M inherits the key properties of the proposal matrix K, such as speed of convergence, while satisfying the prescribed specifications. Thus the choice of matrix K is critical for the performance of the M–H algorithm [14], in order to speed up the *warm up* phase, i.e., the transient regime. However, currently, the M–H algorithm cannot impose hard constraints on the probability distribution vector during the warm up phase, such as upper bound constraints on the probability distribution. These hard constraints on the probability vector of the Markov chain are often referred to as *safety* constraints [15,16].

Safety constraints are critical in applications where the violations during transients can cause a failure of the system. The primary focus of this paper is on swarm control [17–19], where overcrowded regions can increase the risk of collisions in space. Having safe transients is also critical for systems where an exogenous process can push the system out of the stationary regime. Some examples of exogenous processes in swarm control are: addition or removal of agents in and out of the swarm and disaster incidents that cause a group of agents in a region to fail. These incidents push the system out of the stationary regime into a new transient regime, so a safe convergence back to the stationary

☆ A preliminary version of this paper was presented at the ACC 2016 (El Chamie and Açıkmeşe, [1]). This paper extends the results and presentation in El Chamie and Açıkmeşe [1] by providing detailed proofs, more generalized results, and applying the theory to randomized motion planning in multi-agent systems.

* Corresponding author.
E-mail addresses: elchamm@utrc.utc.com (M. El Chamie), behcet@uw.edu (B. Açıkmeşe).

https://doi.org/10.1016/j.sysconle.2017.10.006
0167-6911/

distribution is necessary and can be provided by the safe M–H algorithm proposed in this paper.

In [16,17], we synthesize Markov matrices with safety constraints via convex optimization methods. The approach in [16,17] requires solving an LMI optimization for the construction of these matrices and it is suited for *constant* offline synthesis of Markov matrices. On the other hand, in this paper, we obtain a computationally efficient M–H algorithm to synthesize a *time-varying* Markov matrix. This allows the system to quickly adapt to time-varying desired distribution specifications without recomputing the proposal matrix, which is a very useful property for the swarm control application where this adaptation must happen in real-time. In addition, we present a *robust* version, which handles a larger set of stationary distributions, and a *fast* version, which optimizes the rate of convergence of the Markov chain.

In summary, the main contributions of this paper are: (i) Incorporating safety into the M–H algorithm for a set of stationary probability distributions; (ii) Providing a new linear-programming-based method to synthesize the proposal matrix of the safe M–H algorithm with some robustness properties; (iii) Studying the speed of convergence of the resulting transition matrix of the robust and safe M–H algorithm; (iv) Applying the robust and safe M–H algorithm to a swarm density control problem, with time-varying desired density specifications.

2. Problem formulation

The probability distribution over the states of a Markov chain can be expressed as a probability vector $\mathbf{x}(t) \in \mathbb{R}^m$ with the relation

$$\mathbf{x}(t+1) = M(t)\mathbf{x}(t) \qquad t = 0, 1, 2, \ldots \tag{1}$$

where $M(t)$ is a column stochastic matrix for all time, i.e., $\mathbf{1}^T M(t) = \mathbf{1}^T$ and $M(t) \geq 0$, with $\geq$ being the component-wise inequality. In many applications, it is desired to design the column stochastic matrix $M(t)$ to satisfy some specifications. For example, in swarm control, also referred to as Randomized Motion Planning (RMP) [17,20], $\mathbf{x}(t)$ describes the probability of an agent (e.g., vehicle) to be in a given region and $M(t)$ determines the probability distributions for possible transitions between these regions. In later sections, we will apply our theoretical findings on the RMP problem. In gossiping and wireless sensor networks [21], Eq. (1) describes the dynamics for the evolution of an estimate of a relevant physical quantity as temperature, pressure, etc. In voting models [22], $\mathbf{x}(t)$ determines the preference of a group of people towards a given object of interest (e.g., application, leader, etc.). In consensus protocols, the transition matrix (also called the weight matrix) is designed for the fastest convergence of the consensus among a group of networked agents [23,24].

The probability vector $\mathbf{x}(t)$ characterizes the behavior of the underlying Markov chain, governed by Eq. (1), both during the transient (warm-up) and steady-state phases. During the warm-up phase, the M–H algorithm samples from $\mathbf{x}(t)$, which is biased by the initial distribution $\mathbf{x}(0)$,[1] where $\mathbf{x}(t)$ satisfies some constraints naturally due to the column stochasticity of $M(t)$, such as $\mathbf{x}(t) \geq 0$ and $\mathbf{1}^T\mathbf{x}(t) = 1$ for all $t = 0, 1, \ldots$. There can also be additional constraints characterized by hard *safety* upper bounds on the probability vector, i.e.,

$$\mathbf{x}(t) \leq \mathbf{d} \text{ for all } t \geq 0, \tag{2}$$

where $\mathbf{d} \in \mathbb{R}^m_+$ is a constant non-negative vector. The classical M–H algorithm does not impose the above safety constraints (2) on the probability vector of the resulting Markov chain. This paper presents a new M–H algorithm for Markov chain synthesis method that handles safety constraints while ensuring other specifications such as reversibility, desired stationary (steady-state) distribution, and transitional constraints. The synthesis of the proposed M–H algorithm is based on a numerically tractable linear programming formulation.

3. Formulation of reversible Markov chain synthesis with safety constraints

3.1. Notation

In this paper, small bold letters are used for vectors (e.g., $\mathbf{x}$ whose elements are indicated as $x_1, x_2, \ldots$), and capital letters are used for matrices (e.g., X whose ith row jth column element is denoted by X_{ij}). A *graph* is denoted by $\mathcal{G} = (\mathbf{V}, \mathcal{E})$ where $\mathbf{V} = \{1, \ldots, m\}$ is the set of *vertices* and $\mathcal{E} \subseteq \mathbf{V} \times \mathbf{V}$ is the set of *edges*. We use the pair (j, i) to denote the edge from vertex j to vertex i. We assume that $(i, i) \in \mathcal{E}$ for all $i \in \mathbf{V}$, i.e., $\mathcal{G}$ contains all the self loops. The *adjacency matrix* A of $\mathcal{G} = (\mathbf{V}, \mathcal{E})$ is: $A_{ij} = 1$ if $(j, i) \in \mathcal{E}$ and $A_{ij} = 0$ otherwise. A summary of the notation is given by Table 1. We will consider the following assumption on the graph $\mathcal{G}$:

Assumption 1. $\mathcal{G}$ is undirected and connected.

In an undirected graph, $(j, i) \in \mathcal{E}$ if and only if $(i, j) \in \mathcal{E}$. A graph is connected if for any pair of vertices i and j, there is a path from i to j, i.e., a sequence of edges $(i, i_1), (i_1, i_2), \ldots, (i_p, j)$ starting from vertex i and ending at vertex j. With Assumption 1, the adjacency matrix A is symmetric and irreducible.

3.2. Markov chain specifications

We consider a Markov chain describing the evolution of a discrete probability vector $\mathbf{x}(t) \in \mathbb{P}^m$ given by (1), where $M(t)$ is a column stochastic matrix for all time t. Motivated by Markov chain terminology, $M(t)$ is referred to as the transition matrix where $M_{ij}(t)$ is the probability of transition from a state j to state i at time t.

We first consider the case with a constant transition matrix $M(t) = M$ for all t, and then discuss the time-varying case after introducing the M–H algorithm for the time-invariant desired distributions. Specifically, our objective is to synthesize a transition matrix M such that the resulting Markov chain (1) has the following properties:

1. The desired probability distribution $\mathbf{v} \in \mathbb{P}^m$ is the stationary distribution: $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{v}, \forall\, \mathbf{x}(0) \in \mathbb{P}^m$.
2. Reversibility: $v_i M_{ji} = v_j M_{ij}$, for all $i, j = 1, \ldots, m$.
3. Transition constraints: $M_{ij} = 0$ when $(i, j) \notin \mathcal{E}$, and $M_{ij} > 0$ when $(i, j) \in \mathcal{E}$ (the set of feasible transitions).
4. Safety constraints: $\mathbf{x}(t) \leq \mathbf{d}$ for all $t \geq 0$, and for a given vector $\mathbf{0} \leq \mathbf{d} \leq \mathbf{1}$.

The transition constraints are described by an adjacency matrix characterizing the set of feasible state transitions. The safety constraints bound the probability distribution during both the transients and at the steady-state.

3.3. Convex formulations of the specifications

This section summarizes our results on convex formulations of the above specifications for a time-invariant Markov matrix M. These convex representations are equivalent to the original specifications and they facilitate the formulation of Linear Matrix Inequality (LMI) problems for the synthesis of reversible Markov chains for given steady-state distributions.

[1] The M–H algorithm eventually samples from the limiting distribution $\lim_{t\to\infty}\mathbf{x}(t) = \mathbf{v}$ that is independent of $\mathbf{x}(0)$.