On the Complexity of Computing Maximum Entropy for Markovian Models

Taolue Chen[1] and Tingting Han[2]

1 **Department of Computer Science, Middlesex University London, UK**
2 **Department of Computer Science and Information Systems, Birkbeck, University of London, UK**

Abstract

We investigate the complexity of computing entropy of various Markovian models including Markov Chains (MCs), Interval Markov Chains (IMCs) and Markov Decision Processes (MDPs). We consider both entropy and entropy rate for general MCs, and study two algorithmic questions, i.e., entropy *approximation* problem and entropy *threshold* problem. The former asks for an approximation of the entropy/entropy rate within a given precision, whereas the latter aims to decide whether they exceed a given threshold. We give polynomial-time algorithms for the approximation problem, and show the threshold problem is in $\mathsf{P}^{\mathsf{CH}_3}$ (hence in PSPACE) and in P assuming some number-theoretic conjectures. Furthermore, we study both questions for IMCs and MDPs where we aim to *maximise* the entropy/entropy rate among an infinite family of MCs associated with the given model. We give various conditional decidability results for the threshold problem, and show the approximation problem is solvable in polynomial-time via convex programming.

1998 ACM Subject Classification G.3 Probability and Statistics, D.2.4 Software/Program Verification

Keywords and phrases Markovian Models, Entropy, Complexity, Probabilistic Verification

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2014.571

1 Introduction

Entropy is one of the most fundamental notions in information theory which usually refers to the *Shannon entropy* in this context [16]. In a nutshell, it is the expected value of the information contained in a message. Markovian processes and entropy are related since the introduction of entropy by Shannon. In particular, Shannon defined and studied technically the *entropy rate* of a *discrete-time Markov chain* (henceforth MC in short) with a finite state space, which is one of the main topics of the current paper.

We identify two types of "entropy" defined in literature for MCs. Essentially entropy is a measure of uncertainty in random variables, and MCs, as a stochastic process, are a sequence of random variables. Naturally this view yields two possible definitions, intuitively the "average" and the "sum" of the entropy of the random variables associated with the MC, respectively:

- the classical definition of entropy, dating back to Shannon, typically known as the *entropy rate*. Informally, this is the time density of the *average* information in a stochastic process. *Henceforth, we refer to this definition as entropy rate.*
- the definition given by Biondi *et al* [7], which is the joint entropy of the (infinite) sequence of random variables in a stochastic process. Although being infinite in general, the authors argue that this represents, for instance, the information leakage where the states

34th Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2014).
Editors: Venkatesh Raman and S. P. Suresh; pp. 571–583

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of the MC are the observables of a deterministic program [7]. *Henceforth, we refer to this definition as entropy.*

Formal accounts are given in Section 3. Definitions of entropy of MCs raise algorithmic challenges. One natural question is, given an MC, how to "compute" its entropy? Note that in general, it is *not* a rational (even not an algebraic) number, which prompts the question what computing means exactly. Technically there are (at least) two possible interpretations which we formulate as the *entropy approximation problem* and the *entropy threshold problem*, respectively. Let $\mathcal{D}$ be an MC and $\hbar$ denote the entropy/entropy rate of $\mathcal{D}$.

- The entropy approximation problem aims to compute, given the error bound $\epsilon > 0$, a rational number θ such that $|\hbar - \theta| \leq \epsilon$;
- The entropy threshold problem aims to decide, given the rational number θ, whether $\hbar \bowtie \theta$, where $\bowtie \in \{<, \leq, =, \geq, >\}$.

Observe that general speaking the approximation problem is no harder than the threshold problem, since it can be solved by a simple binary search with the threshold problem as the oracle. However, the converse does *not* hold in general.

On top of a purely probabilistic model like MCs, it is probably more interesting to consider probabilistic models with *nondeterminism*, typically Interval Markov chains (IMCs) and Markov Decision Processes (MDPs). MDPs [26] are a well-established model which is widely used in, for instance, robotics, automated control, economics, and manufacturing. IMCs [22] are MCs where each transition probability is assumed to be within a range (interval). They are introduced to faithfully capture the scenario where transition probabilities are usually estimated by statistical experiments and thus it is not realistic to assume they are exact.

By and large, a probabilistic model with nondeterminism usually denotes an (infinite) family of pure probabilistic models. Among these models, selecting the one with the *maximum* entropy is one of the central questions in information theory [16]. As before, it raises algorithmic challenges as well, i.e., given an IMC or MDP which denotes an infinite family of MCs, how to "compute" the *maximum entropy*? Note the dichotomy of the approximation and the threshold problem exists here as well, which we shall refer to the *maximum entropy approximation problem* and the *maximum entropy threshold problem*, respectively.

Entropy of probabilistic models has a wide range of applications, in particular in security [12, 6, 30]. As a concrete example which is one of the motivations of the current paper, in a recent paper [7], all possible attacks to a system are encoded as an IMC, and the channel capacity computation reduces to finding an MC with highest entropy. Note that tool support has been already available [8].

Contributions. In this paper we are mainly interested in the algorithmic aspects of entropy for Markovian models. In particular, we carry out a theoretical study on the complexity of computing (maximum) entropy for MCs, IMCs, and MDPs. The main contributions are summarised as follows:

1. We consider the definition of entropy rate for general (not ergodic) MCs, and give a characterisation in terms of local entropy;
2. We identify the complexity of the entropy approximation problem and the entropy threshold problem for MCs;
3. We identify the complexity of the approximation problem for maximum entropy/entropy rate for IMCs, and we obtain *conditional* decidability for the threshold problem. These results can be adapted to the MDP model as well.

The main results of the paper are summarised in Table 1.

Table 1 Complexity of computing entropy/entropy rate

	approximation	threshold
MC	P	$\mathsf{P}^{\mathsf{CH}_3}$ (conditional in P)
IMC/MDP	P	conditional decidable

Some remarks are in order:

- Regarding **1**, in literature entropy rate is defined exclusively over *irreducible* (sometimes called ergodic) MCs where the celebrated Shannon-McMillan-Breiman theorem [16] actually gives a characterisation in terms of stationary distribution and local entropy. However, for computer science applications, MC models are seldom irreducible. Hence we provide a characterisation for general (finite-state) MCs, inspired by the one in [7].
- For the "computation" of entropy of MCs, [7] states that it can be done in polynomial time. Although not stated explicitly, this actually refers to the approximation problem. The threshold problem is not addressed in [7], nor the corresponding problems wrt. the entropy rate.
- For the "computation" of maximum entropy of IMCs, [7] considers the approximation problem. The authors reduce the problem to non-linear programming (over a convex polytope though) to which no complexity result is given. Here, instead, we show, by reducing to *convex programming*, the approximation problem can be solved in polynomial time. Note that the formulation in [7] is not convex in general, so we cannot start from there straightforwardly.
- For maximisation of entropy rate, it is actually a classical topic for MCs and semi-MCs. A classical result, due to Parry [24], shows how to define a (stationary) MC (called Shannon-Parry MC) over a given strongly connected graph to achieve the maximum entropy rate. More recent results focus on finding a (semi-)MC with the maximum entropy rate when its stationary distribution is constrained in certain ways, see, e.g., [19]. In contrast, here we work on the entropy rate for general IMCs and MDPs. To the best of our knowledge this is the first work of this type.

Related work. Apart from the work we have discussed before, [30, 12] studied the complexity of quantitative information flow for boolean and recursive programs, whereas [11] studied the information-leakage bounding problem (wrt. Shannon entropy) for deterministic transition systems. [9] studied entropy in process algebra. These models and questions are considerably different from ours. [13, 27, 15, 25, 4] studied IMCs and their model checking problems. The technique to solve convex programming is inspired by [25]. We also mention that [2] generalised Parry's result to the graph generated by timed automata.

An extended version of the paper [14] contains proofs, detailed expositions, and in particular, all results for MDPs.

2 Preliminaries

Let $\mathbb{N}, \mathbb{Q}, \mathbb{R}$ denote the set of natural, rational, real numbers, respectively. Given any finite set S, we write $\Delta(S)$ for the set of *probabilistic distributions* over S, i.e., functions $\mu : S \to [0, 1]$ with $\sum_{s \in S} \mu(s) = 1$. For any vector $\vec{x}$, we write $\vec{x}_i$ for the entry of $\vec{x}$ corresponding to the index i, and $\vec{x} \geq 0$ if $\vec{x}_i \geq 0$ for each i. Throughout this paper, $X, Y, \cdots$ denote discrete *random variables* (RVs), usually over a finite set of outcomes. For the RV X, we often denote

the set of outcomes as $\mathcal{X} = \{x_1, \cdots, x_n\}$ which is ranged over by x. In this context, we also write $\Pr(X = x)$ or simply $p(x)$ for the *probability mass function*.

2.1 (Interval) DTMCs

▶ **Definition 1** (MC). A *(discrete-time) Markov chain* (MC) is a tuple $\mathcal{D} = (S, \alpha, \mathbf{P})$, where S is a finite set of *states*; $\alpha \in \Delta(S)$ is the *initial distribution*; and $\mathbf{P} : S \times S \to [0, 1]$ is the *transition probability matrix*, satisfying $\forall s \in S$, $\sum_{s' \in S} \mathbf{P}(s, s') = 1$.

Alternatively, an MC can be defined as a stochastic process $\{X_n\}_{n \geq 0}$, where each X_n is a discrete RV over S. The process respects the Markov property, i.e., $\Pr(X_n = s_n | X_{n-1} = s_{n-1}, \cdots, X_0 = s_0) = \Pr(X_n = s_n | X_{n-1} = s_{n-1}) = \mathbf{P}(s_{n-1}, s_n)$ for any $s_0, s_1, \cdots, s_n \in S$ and $n \in \mathbb{N}$. Note that $\Pr(X_n = s)$ denotes the probability of being in state s at time n. The *transient distribution* of $\mathcal{D}$ is denoted by $\pi^{(n)} \in \Delta(S)$, which can be computed by $\pi^{(n)} = \alpha \mathbf{P}^n$. It is known that $\Pr(X_n = s) = \pi^{(n)}_s$.

For a finite MC, we often use graph-theoretical notations which refer to the underlying digraph of $\mathcal{D}$. Essentially the vertices of the digraph are states of $\mathcal{D}$, and there is an edge from s to t iff $\mathbf{P}(s, t) > 0$. The following notions are standard.

▶ **Definition 2.**
- A subset $T \subseteq S$ is *strongly connected* if for each pair of states $s, t \in T$, t is reachable from s. A *strongly connected component* (SCC) T of an MC $\mathcal{D}$ denotes a strongly connected set of states such that no proper superset of T is strongly connected.
- A *bottom strongly connected component* (BSCC) T is an SCC from which no state outside T is reachable.

We write $\mathcal{E}(\mathcal{D})$ for the set of all SCCs of $\mathcal{D}$ and $\mathcal{B}(\mathcal{D}) \subseteq \mathcal{E}(\mathcal{D})$ for the set of all BSCCs of $\mathcal{D}$.

▶ **Definition 3.**
- A state s is *absorbing* if $\mathbf{P}(s, s) = 1$, i.e. s contains only a self-loop. An MC is *absorbing* if every state can reach an absorbing state.
- A state s is *transient* if, starting in state s, there is a non-zero probability that it will never return to s; otherwise s is *recurrent*.
- A state s is *deterministic* if the distribution $\mathbf{P}(s, \cdot)$ is Dirac, i.e. there is a unique t such that $\mathbf{P}(s, t) = 1$; otherwise s is *stochastic*.
- An MC is *irreducible* if its underlying digraph is strongly connected.

▶ **Definition 4** (IMC). An *interval-valued (discrete-time) Markov chain* (IMC) is a tuple $\mathcal{I} = (S, \alpha, \mathbf{P}^l, \mathbf{P}^u)$, where S, α are defined as in Definition 1; $\mathbf{P}^l, \mathbf{P}^u : S \times S \to [0, 1]$ are two transition probability matrices, where $\mathbf{P}^l(s, s')$ (resp. $\mathbf{P}^u(s, s')$) gives the *lower* (resp. *upper*) bound of the transition probability from state s to s'.

Semantics. There are two semantic interpretations of IMCs [27], i.e., *Uncertain Markov Chains* (UMC) and *Interval Markov Decision Processes* (IMDP). In this paper, following [7], we mainly focus on the UMC semantics. An IMC $\mathcal{I} = (S, \alpha, \mathbf{P}^l, \mathbf{P}^u)$ represents an infinite set of MCs, denoted by $[\mathcal{I}]$, where for each MC $(S, \alpha, \mathbf{P}) \in [\mathcal{I}]$, $\mathbf{P}^l(s, s') \leq \mathbf{P}(s, s') \leq \mathbf{P}^u(s, s')$ for all pairs of states $s, s' \in S$. Intuitively, under this semantics we assume that the external environment nondeterministically selects an MC from the set $[\mathcal{I}]$ at the beginning and then all the transitions take place according to the chosen MC. Without loss of generality, *we only consider IMC $\mathcal{I}$ with* $[\mathcal{I}] \neq \emptyset$, i.e., there exists at least one implementation. This condition can be easily checked.

Similar to MCs, we can also view an IMC as a digraph such that there is an edge from s to t iff $\mathbf{P}^u(s, t) > 0$. In this way, we can speak of the set of all SCCs and BSCCs of an IMC $\mathcal{I}$ which we denote by $\mathcal{E}(\mathcal{I})$ and $\mathcal{B}(\mathcal{I})$, respectively.

For complexity consideration, for the introduced probabilistic models, we assume that all the probabilities are rational numbers. We define the size of $\mathcal{D}$ (resp. $\mathcal{I}$), denoted by $\sharp\mathcal{D}$ (resp. $\sharp\mathcal{I}$), as the size of the representation of $\mathcal{D}$ (resp. $\mathcal{I}$). Here rational numbers (probabilities) are represented as quotients of integers written in binary. The size of a rational number is the sum of the bit lengths of its numerator and denominator and the size of a matrix is the sum of the sizes of its entries. When stating a complexity result, we assume the standard Turing model.

2.2 Information theory

For a RV X with outcomes $\{x_1, \cdots, x_n\}$, the *Shannon entropy* of X is defined as $\mathbb{H}(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$. (Note that by convention we define $0 \log 0 = 0$ as $\lim_{x\to 0} x \log x = 0$). All logarithms are to the base 2; however our results are *independent* of the base. The definition of Shannon entropy is easily generalised to *joint entropy*, the entropy of several RVs computed jointly. Namely $\mathbb{H}(X_1, \cdots, X_n) = -\sum_{x_1\in\mathcal{X}_1} \cdots \sum_{x_n\in\mathcal{X}_n} p(x_1, \cdots, x_n) \log p(x_1, \cdots, x_n)$. We also define *conditional entropy* which quantifies the amount of information needed to describe the outcome of a random variable Y given that the value of another random variable X is known. Namely $\mathbb{H}(Y|X) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x, y) \log \frac{p(x)}{p(x, y)}$. The *chain rule* relates the joint entropy and the conditional entropy, namely, $\mathbb{H}(Y|X) = \mathbb{H}(X, Y) - \mathbb{H}(X)$. It follows that the joint entropy can be calculated using conditional entropy, i.e., $\mathbb{H}(X_0, \cdots, X_n) = \mathbb{H}(X_0) + \mathbb{H}(X_1|X_0) + \cdots + \mathbb{H}(X_n|X_1, \cdots, X_{n-1})$.

3 Entropy of MCs

In this section, we define and characterise the entropy/entropy rate for an MC which we fix to be $\mathcal{D} = (S, \alpha, \mathbf{P})$. $\mathcal{D}$ is equipped with a stochastic process as $\{X_n\}_{n\in\mathbb{N}}$. Let's start from a basic property which can be deduced from the memoryless property.

▶ **Lemma 5.** $\mathbb{H}(X_n|X_1, \cdots, X_{n-1}) = \mathbb{H}(X_n|X_{n-1})$.

It turns out that the notion of *local entropy* [7] plays a central role in developing a characterisation of entropy/entropy rate for MCs which are amenable to computation.

▶ **Definition 6** ([7])**.** For any given MC $\mathcal{D}$ and state $s \in S$, the *local entropy* $L(s)$ is defined as $\mathbb{H}(\mathbf{P}(s, \cdot))$, i.e, $-\sum_{t\in S} \mathbf{P}(s, t) \log \mathbf{P}(s, t)$.

3.1 Entropy for absorbing MCs

▶ **Definition 7** ([7])**.** Given an MC $\mathcal{D}$, the entropy of $\mathcal{D}$, denoted $H(\mathcal{D})$, is defined as $H(\mathcal{D}) = \mathbb{H}(X_0, X_1, \cdots)$.

We note that [7] also provides an elegant characterisation. Define $\xi(s) = \sum_{n=0}^{\infty} \pi_s^{(n)}$. (It is called residence time in [7].) Note that basic theory of MCs implies that the state s is *recurrent* if $\xi(s) = \infty$, and is *transient* iff $\xi(s) < \infty$. We write $\vec{\xi}$ for the vector $(\xi(s))_{s\in S}$.

▶ **Theorem 8.** *$H(\mathcal{D}) = \sum_{s\in S} L(s)\xi(s) + \mathbb{H}(\alpha)$, where $\mathbb{H}(\alpha) = -\sum_{s\in S} \alpha(s) \log \alpha(s)$.*

▶ Remark. [7] defines the entropy for general MCs whereas here we assume MCs are absorbing. This does not lose any generality. Mostly we are only interested in MCs with finite entropy, and one easily observes: $H(\mathcal{D})$ is finite *iff* the local entropy of each recurrent state is 0. Note

that *absorbing* MCs admits that each recurrent state is made absorbing and thus has local entropy 0.

We also note there is slight difference on $\mathbb{H}(\alpha)$ between our version and that of [7] in Theorem 8. The paper [7] assumes a unique initial state in MCs (i.e., α is Dirac) where $\mathbb{H}(\alpha) = 0$; here we assume a (slightly more) general initial distribution α.

3.2 Entropy rate for general MCs

In contrast to the entropy, the *entropy rate* is defined as

▶ **Definition 9.** Given an MC $\mathcal{D}$, the entropy rate of $\mathcal{D}$, denoted $\nabla H(\mathcal{D})$ is defined as

$$\nabla H(\mathcal{D}) = \lim_{n\to\infty} \frac{1}{n}\mathbb{H}(X_0, \cdots, X_n)$$

As before we characterise $\nabla H(\mathcal{D})$ by local entropy. Define $\zeta(s) = \lim_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \pi_s^{(i)}$ and write $\vec{\zeta}$ for the vector $(\zeta(s))_{s\in S}$. We have the following result:

▶ **Theorem 10.** $\nabla H(\mathcal{D}) = \sum_{s\in S} L(s)\zeta(s)$.

▶ Remark. Typically in literature (e.g. [16, 19]), the entropy rate is defined only for an ergodic MC. In that case, one has $\nabla H'(\mathcal{D}) = \lim_{n\to\infty} \mathbb{H}(X_n \mid X_1, \cdots, X_{n-1})$. For ergodic MCs (more generally all stationary processes where MCs are a special case), these two quantities coincide and by Lemma 5, the entropy rate is given by $\nabla H'(\mathcal{D}) = \lim_{n\to\infty} \mathbb{H}(X_n \mid X_{n-1})$.

4 Computing entropy in MCs

In this section, we will focus on the *entropy threshold* problem which asks: given an MC $\mathcal{D}$ and $\theta \in \mathbb{Q}$, does $H(\mathcal{D}) \bowtie \theta$ hold for $\bowtie \in \{\leq, <, =, >, \geq\}$? We assume some familiarity with *straight-line programs* and the *counting hierarchy* (cf. [1] or [14]). In particular, the problem *PosSLP* is to decide, given a straight-line program, whether the integer it represents is *positive*. PosSLP belongs to the complexity class $\mathsf{P}^{\mathsf{CH}_3}$ and thus to the fourth-level of the counting hierarchy [1]. We note that counting hierarchy is contained in PSPACE, but it is unlikely to be complete to PSPACE. The following propositions are slight generalisations of [12] and [18], respectively.

▶ **Proposition 11.** *Given $p_1, \cdots, p_n, q_1, \cdots, q_n, \theta \in \mathbb{Q}$, deciding whether $\sum_{i=1}^{n} p_i \log q_i \bowtie \theta$ for $\bowtie \in \{\leq, <, >, \geq\}$ reduces to PosSLP in polynomial time.*

▶ **Proposition 12.** *Given $p_1, \cdots, p_n, q_1, \cdots, q_n, \theta \in \mathbb{Q}$, $\sum_{i=1}^{n} p_i \log q_i = \theta$ is decidable in polynomial time.*

ABC/Lang-Waldschmidt conjecture implies P. An interesting question is whether one could obtain a lower-bound. This is left as an open question, but the following result somehow discourages such efforts. Indeed, the following proposition can be easily obtained by essentially [18, Proposition 3.7(1)].

▶ **Proposition 13.** *Assume $p_1, \cdots, p_n, q_1, \cdots, q_n, \theta \in \mathbb{Q}$. If the* ABC conjecture *holds, or if the* Lang-Waldschmidt *conjecture holds, then $\sum_{i=1}^{n} p_i \log q_i \bowtie \theta$ for $\bowtie \in \{\leq, <, >, \geq\}$ is decidable in polynomial time.*

Note that the ABC and the Lang-Waldschmidt conjecture (cf. [18] for precise formulations and reference therein) are conjectures in transcendence theory which are widely believed to be true. (For instance, in 2012 there was an announced proof of the ABC conjecture by S. Mochizuki.)

Below we apply these results to the entropy threshold problem of MCs.

4.1 Entropy

Owing to Theorem 8, computing $H(\mathcal{D})$ reduces to computing $\vec{\xi}$. In [7] it is stated that ξ can be computed in polynomial time. Here we need to elaborate this claim to obtain complexity results. This is rather straightforward. For a given absorbing MC which has t transient states and r absorbing states, the transition probability matrix $\mathbf{P}$ can be written as $\mathbf{P} = \begin{bmatrix} Q & R \\ 0 & \mathbf{I}_r \end{bmatrix}$, where Q is a $t \times t$ matrix, R is a nonzero $t \times r$ matrix, and $\mathbf{I}_r$ is an $r \times r$ identity matrix. A basic property of absorbing MCs is that the *fundamental matrix* $\mathbf{I} - Q$ is invertible [21], and we have the following:

▶ **Proposition 14** ([21]). *For absorbing MC, $\vec{\xi} = \alpha'(\mathbf{I} - Q)^{-1}$ where α' is the restriction of α to the t transient states.*

Basic linear algebra reveals that $\vec{\xi}$ can be computed in cubic-time via, e.g., Gauss elimination, and the size of $\vec{\xi}$ is polynomially bounded by $\sharp\mathcal{D}$ (see, e.g., [20]). It then follows from Proposition 11 and Proposition 12 that:

▶ **Theorem 15.** *Given an MC $\mathcal{D}$,*

- *Deciding $H(\mathcal{D}) \bowtie \theta$ for $\bowtie \in \{<, \leq, \geq, >\}$ is in $\mathsf{P}^{\mathsf{CH}_3}$, and is in P assuming the ABC or the Lang-Waldschmidt conjecture.*
- *Deciding $H(\mathcal{D}) = \theta$ is in P.*

4.2 Entropy rate

Owing to Theorem 10, computing $\nabla H(\mathcal{D})$ reduces to computing $\vec{\zeta}$. For (finite) irreducible MC, $\vec{\zeta}$ coincides to the *stationary distribution* π which is unique and independent of the initial distribution. In this case, Theorem 10 yields that $\nabla H(\mathcal{D}) = \sum_{s \in S} L(s)\pi(s)$, which is exactly the classical result, see, e.g., [16]. For general MCs, the transition probability matrix $\mathbf{P}$ has the form

$$\mathbf{P} = \begin{pmatrix} Q & R_1 & R_2 & \cdots & R_h \\ \mathbf{0} & B_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & B_h \end{pmatrix}$$

where Q corresponds to transient states, and B_i $(1 \leq i \leq h)$ corresponds to the BSCCs (recurrent states).

▶ **Proposition 16.** *For any MC,*

$$\vec{\zeta} = \alpha \cdot \begin{pmatrix} \mathbf{0} & (\mathbf{I} - Q)^{-1} R_1 \mathbf{1}^{\mathrm{T}} \vec{y}_1 & (\mathbf{I} - Q)^{-1} R_2 \mathbf{1}^{\mathrm{T}} \vec{y}_2 & \cdots & (\mathbf{I} - Q)^{-1} R_h \mathbf{1}^{\mathrm{T}} \vec{y}_h \\ \mathbf{0} & \mathbf{1}^{\mathrm{T}} \vec{y}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}^{\mathrm{T}} \vec{y}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}^{\mathrm{T}} \vec{y}_h \end{pmatrix}$$

where $\vec{y}_i$ *is the solution of the system of linear equations:*

$$\vec{y}_i B_i = \vec{y}_i \text{ and } \mathbf{1}\vec{y} = 1$$

and $\mathbf{1} = (1, \cdots, 1)$.

Similar to the previous section, the size of $\vec{\zeta}$ is polynomially bounded by $\sharp\mathcal{D}$. It then follows from Proposition 11 and Proposition 12 that:

▶ **Theorem 17.** *Given an MC* $\mathcal{D}$*,*

- *Deciding* $\nabla H(\mathcal{D}) \bowtie \theta$ *for* $\bowtie \in \{<, \leq, \geq, >\}$ *is in* $\mathsf{P}^{\mathsf{CH}_3}$*, and is in* P *assuming the ABC or the Lang-Waldschmidt conjecture.*
- *Deciding* $\nabla H(\mathcal{D}) = \theta$ *is in* P*.*

4.3 Approximation problems

To complete the picture, we show that one can easily approximate $\sum_{i=1}^{n} p_i \log q_i$ up to a given error bound ϵ in polynomial time.

Let $N = n \cdot \max_{1 \leq i \leq n} |p_i|$. For each $1 \leq i \leq n$, we can compute $\theta_i \in \mathbb{Q}$ in polynomial-time [10, 18] such that $|\log q_i - \theta_i| < \frac{\epsilon}{N}$ (note that the size of N is bounded polynomially by the size of the input). Observe that

$$\left| \sum_{i=1}^{n} p_i \log q_i - \sum_{i=1}^{n} p_i \theta_i \right| \leq \left| \sum_{i=1}^{n} p_i (\log q_i - \theta_i) \right| \leq \sum_{i=1}^{n} |p_i| \frac{\epsilon}{N} \leq \epsilon.$$

Hence $\sum_{i=1}^{n} p_i \theta_i$, which can be computed in polynomial-time, is an approximation of $\sum_{i=1}^{n} p_i \log q_i$ up to ϵ. Note that, however, unfortunately this does *not* yield an efficient decision procedure for $\sum_{i=1}^{n} p_i \log q_i \bowtie \theta$. It follows that

▶ **Theorem 18.** *Given an MC* $\mathcal{D}$ *and* $\epsilon > 0$*, both* $H(\mathcal{D})$ *and* $\nabla H(\mathcal{D})$ *can be approximated up to* ϵ *in polynomial-time in* $\sharp\mathcal{D}$ *and* $\log(\frac{1}{\epsilon})$*.*

(Note that this result for entropy is implied in [7] without proof.)

5 Computing the maximum entropy in IMCs

In this section, we turn our attention to IMCs. Recall that an IMC $\mathcal{I}$ represents a set of MCs $[\mathcal{I}]$. We are interested in maximising the entropy/entropy rate of $\mathcal{I}$. The formal definitions are given as follows:

▶ **Definition 19.** Given an IMC $\mathcal{I}$,

- the *maximum entropy* of $\mathcal{I}$, $\overline{H}(\mathcal{I})$, is defined as $\overline{H}(\mathcal{I}) = \sup\{H(\mathcal{D}) \mid \mathcal{D} \in [\mathcal{I}]\}$;
- the *maximum entropy rate* of $\mathcal{I}$, $\overline{\nabla H}(\mathcal{I})$, is defined as $\overline{\nabla H}(\mathcal{I}) = \sup\{\nabla H(\mathcal{D}) \mid \mathcal{D} \in [\mathcal{I}]\}$.

Below we focus on the computation of maximum entropy/entropy rate. In contrast to the previous section, we mainly concentrate on the approximation problem. Results regarding the threshold problem are presented in Section 5.3, though. Throughout this section, we fix an IMC $\mathcal{I} = (S, \alpha, \mathbf{P}^l, \mathbf{P}^u)$.

5.1 Entropy

As pointed out by [7], it could be the case that $\overline{H}(\mathcal{I}) = \infty$ even if for all $\mathcal{D} \in [\mathcal{I}]$, $H(\mathcal{D}) < \infty$. To tackle this issue, an algorithm is given there to determine whether $\overline{H}(\mathcal{I}) = \infty$. In light of this, we *assume that* $\overline{H}(\mathcal{I}) < \infty$. One sufficient condition to guarantee finite maximum entropy is to impose that for any states s and t, $\mathbf{P}^u(s,t) > 0$ implies $\mathbf{P}^l(s,t) > 0$. This is actually a mild assumption in practice (for instance, see [7], Fig. 5). Note that it is also a (lightweight) syntactic way to impose the *Positive* UMC semantics [13].

For $\mathcal{I}$ with $\overline{H}(\mathcal{I}) < \infty$, it cannot be the case that a state is recurrent in some implementation and stochastic in another implementation [7]. Namely, if a state is recurrent in some implementation, it must be deterministic in all implementations, and thus is made absorbing. We denote by $G \subseteq S$ the set of states which are recurrent in *some* implementation of $\mathcal{I}$; G is easily identified by the algorithm in [7].

For each state $s \in S \setminus G$, we introduce a vector of variables $\vec{x}_s = (x_{s,t})_{t \in S}$, and a vector of variables $\vec{y} = (y_s)_{s \in S}$. We define $\Omega(s)$ to be a set of vectors as:

$$\vec{x}_s \in \Omega(s) \;\text{ iff }\; \begin{cases} \sum_{t \in S} x_{s,t} = 1 \\ \mathbf{P}^l(s,t) \le x_{s,t} \le \mathbf{P}^u(s,t), \text{ for each } t \in S \end{cases} \tag{1}$$

(Note that here we abuse the notation slightly by identifying variables and *valuations* of the variables.) For simplicity, we define, for $\vec{x}_s$ and $\vec{y}$,

$$\Gamma(\vec{x}_s, \vec{y}) = \sum_{t \in S} x_{s,t} y_t - \sum_{t \in S} x_{s,t} \log x_{s,t} \ . \tag{2}$$

We then consider the following non-linear program over $\vec{x}_s$ for all $s \in S \setminus G$ and $\vec{y}$:

$$\begin{aligned} \text{minimise} \quad & \sum_{s \in S \setminus G} \alpha(s) y_s \\ \text{subject to} \quad & y_s \ge \max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y}) \quad && s \notin G \\ & y_s = 0 && s \in G \end{aligned} \tag{3}$$

▶ **Proposition 20.** *The optimal value of* (3) *is equal to* $\overline{H}(\mathcal{I}) - \mathbb{H}(\alpha)$.

We remark that (3) is reminiscent of the *expected total reward* objective (or the stochastic shortest path problem) for MDPs [26, 17, 5]. This does not come in surprise in light of Theorem 8, which might give some intuition underlying (3); cf. [14].

Nevertheless it remains to solve (3). This is rather involved and we only give a rough sketch here. Observe that we have a nested optimisation problem because of the presence of an inner optimisation $\max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y})$ in (3). The main strategy is to apply the Lagrange duality to replace it by some "min" (see $\widetilde{\Gamma}$ below). We introduce, apart from $\vec{y}$, variables $\vec{\lambda}^l_s = (\lambda^l_{s,t})_{t \in S}$, $\vec{\lambda}^u_s = (\lambda^u_{s,t})_{t \in S}$ and ν_s for each $s \in S \setminus G$.

It can be shown that (3) is equivalent to

$$\begin{aligned} \text{minimise} \quad & \sum_{s \in S \setminus G} \alpha(s) y_s \\ \text{subject to} \quad & y_s \ge \widetilde{\Gamma}(\vec{\lambda}_s, \nu_s, \vec{y}) && s \notin G \\ & y_s = 0 && s \in G \\ & \lambda^l_{s,t} \ge 0, \lambda^u_{s,t}, \nu_s \ge 0 && s \notin G, t \in S \end{aligned} \tag{4}$$

where $\widetilde{\Gamma}(\vec{\lambda}_s, \nu_s, \vec{y}) = -\vec{b}_s^{\mathrm{T}}\vec{\lambda}_s^u + \vec{a}_s^{\mathrm{T}}\vec{\lambda}_s^l - \nu_s + e^{-1}\log e \cdot 2^{\nu_s} \cdot (\sum_{t\in S} 2^{\vec{\lambda}_{s,t}^u - \vec{\lambda}_{s,t}^l + y_t})$ and $\vec{a}_s = (\mathbf{P}^l(s,t))_{t\in S}$ and $\vec{b}_s = (\mathbf{P}^u(s,t))_{t\in S}$. (Note that log is to base 2.)

It turns out that (4) is a convex program which can be solved by, e.g., the ellipsoid algorithm or interior-point methods in polynomial time [3, 20]. We obtain

▶ **Theorem 21.** *Given an IMC $\mathcal{I}$ and $\epsilon > 0$, $\overline{H}(\mathcal{I})$ can be approximated upper to ϵ in polynomial-time in $\sharp\mathcal{I}$ and $\log(\frac{1}{\epsilon})$.*

5.2 Entropy rate

In this section, we study the approximation problem for $\overline{\nabla H}(\mathcal{I})$. Firstly we assert that $\overline{\nabla H}(\mathcal{I}) < \infty$ (cf. [14]).

Recall $\mathcal{E}(\mathcal{I})$ is the set of SCCs of $\mathcal{I}$. For each SCC $B \in \mathcal{E}(\mathcal{I})$, we introduce a variable r, a vector of variables $\vec{y} = (y_s)_{s\in B}$, and for each $s \in B$, a vector of variables $\vec{x}_s = (x_{s,t})_{t\in S}$. Recall that $\Omega(s)$ and $\Gamma(\vec{x}_s, \vec{y})$ are defined as in (1) and (2), respectively. We consider the following non-linear program:

$$\begin{aligned} &\text{minimise} && r \\ &\text{subject to} && r + y_s \geq \max_{\vec{x}_s\in\Omega(s)} \Gamma(\vec{x}_s, \vec{y}) \quad s \in B \end{aligned} \tag{5}$$

For each B, we obtain r_B as the optimal value of (5). Note that each state s must belong to a unique $B \in \mathcal{E}(\mathcal{I})$. For simplicity, we define, for a given vector $(z_s)_{s\in S}$, $\Lambda(\vec{x}_s, \vec{z}) = \sum_{t\in S} x_{s,t}\cdot z_t$. We then consider the following non-linear program

$$\begin{aligned} &\text{minimise} && \sum_{s\in S} \alpha(s) z_s \\ &\text{subject to} && z_s \geq \max_{\vec{x}_s\in\Omega(s)} \Lambda(\vec{x}_s, \vec{z}) && s \in S \\ & && z_s \geq r_B && s \in S \text{ and } s \in B \end{aligned} \tag{6}$$

▶ **Proposition 22.** *$\overline{\nabla H}(\mathcal{I})$ is equal to the optimal value of (6) (which depends on (5)).*

As before, we remark that (6) and (5) are reminiscent of the *limiting average reward* objective for MDPs [26, 5]. This does not come in surprise in light of Theorem 10, which might give some intuition; cf. also [14].

It remains to solve (5) and (6). In the same vein as in Section 5.1, for each B we can *approximate* r_B by some $\theta_B \in \mathbb{Q}$ upper to the given $\epsilon > 0$. We then substitute (6) for each θ_B, and solve the resulting program. It remains to show that (6) does not "propagate" the error introduced in θ_B as it is merely an approximation of the real value r_B. To this end, observe that the optimal value of (6) can be regarded as a function g over $\vec{r} = (r_B)_{B\in\mathcal{E}(\mathcal{I})}$. We have the following result showing the value of (6) is bounded by the "perturbation" of its parameters r_B's. (Note that $\|\cdot\|$ denotes the ∞-norm for vectors.)

▶ **Proposition 23.** *If $\|\vec{r} - \vec{r}'\| \leq \epsilon$, then $|g(\vec{r}) - g(\vec{r}')| \leq \epsilon$.*

We conclude that

▶ **Theorem 24.** *Given an IMC $\mathcal{I}$ and $\epsilon > 0$, $\overline{\nabla H}(\mathcal{I})$ can be approximated upper to ϵ in polynomial-time in $\sharp\mathcal{I}$ and $\log(\frac{1}{\epsilon})$.*

5.3 Threshold problem

In this section, we focus on the maximum entropy/entropy rate *threshold* problem, namely, to decide whether $\overline{H}(\mathcal{I}) \bowtie \theta$ or $\overline{\nabla H}(\mathcal{I}) \bowtie \theta$ for a given $\theta \in \mathbb{Q}$. Recall that we assume $\overline{H}(\mathcal{I}) < \infty$ otherwise the problem is trivial. Below we present two *conditional* decidability results; the unconditional decidability is left as an open problem. We mainly present the results for $\overline{H}(\mathcal{I})$ and the case $\bowtie = \geq$. Other cases can be derived in a similar way and can be found in the full version [14].

By first-order theory. It turns out deciding $\overline{H}(\mathcal{I}) \geq \theta$ amounts to checking

$$\exists \vec{x}, \vec{y}. \bigwedge \begin{cases} \sum_{s \in S \setminus G} \alpha(s) y_s \geq \theta \\ y_s = \sum_{t \in S} x_{s,t} y_t - \sum_{t \in S} x_{s,t} \log x_{s,t} \quad \forall s \in S \setminus G \\ y_s = 0 \quad \forall s \in G \\ \mathbf{P}^l(s,t) \leq x_{s,t} \leq \mathbf{P}^u(s,t) \quad \forall s \in S \setminus G, t \in S \\ \sum_{t \in S} x_{s,t} = 1 \quad \forall s \in S \setminus G \end{cases}$$

where $\vec{x}$ is the concatenation of $\vec{x}_s = (x_{s,t})_{t \in S}$ for $s \in S \setminus G$ and $\vec{y} = (y_s)_{s \in S}$. Recall that G is the set of states which are recurrent in some implementation of $\mathcal{I}$. Evidently this is a formula in the first-order theory of ordered real fields *extended with exponential functions* $(\mathbb{R}, +, -, \cdot, e^x, 0, 1, \leq)$. The theory is known to be o-minimal by the celebrated Wilkie's theorem [29]. However, its decidability is a long-standing open problem in model theory, known as *Tarski's exponential function problem*. A notable result by Macintyre and Wilkie [23] asserts that it is decidable provided the Schanuel's conjecture in transcendence theory is true (which is widely believed to be the case; in fact only a (weaker) real version of the conjecture is needed.) Hence, we obtain a conditional decidability for the maximum entropy threshold problem of IMCs. Note that it is high unlikely that the problem is undecidable, because it would refute the Schanuel's conjecture.

By non-singularity assumption. We can obtain the decidability of the maximum entropy threshold problem by assuming that $\overline{H}(\mathcal{I}) \neq \theta$. To see this, one can simply compute a sequence of approximations of $\overline{H}(\mathcal{I})$ by the approach in Section 5.1, i.e., $\hbar_n$ with $|\overline{H}(\mathcal{I}) - \hbar_n| \leq \frac{1}{2^n}$. The procedure stops when $\hbar_n - \frac{1}{2^n} - \theta$ and $\hbar_n + \frac{1}{2^n} - \theta$ have the same sign. Then $\overline{H}(\mathcal{I}) > \theta$ iff $\hbar_n - \frac{1}{2^n} > \theta$ (or equivalently $\hbar_n + \frac{1}{2^n} > \theta$). Note that we assume $\overline{H}(\mathcal{I}) \neq \theta$, so n must exist as one can take $n = \lceil \log(\frac{1}{|\overline{H}(\mathcal{I}) - \theta|}) \rceil$ although n is not bounded *a priori*.

We conclude this section by the following theorem:

▶ **Theorem 25.** *Given an IMC $\mathcal{I}$. We have that*

- *if the first-order theory of $(\mathbb{R}, +, -, \cdot, e^x, 0, 1, \leq)$ is decidable (which is implied by Schanuel's conjecture), then $\overline{H}(\mathcal{I}) \bowtie \theta$ and $\overline{\nabla H}(\mathcal{I}) \bowtie \theta$ are decidable for $\bowtie \in \{\leq, <, =, >, \geq\}$;*
- *if $\overline{H}(\mathcal{I}) \neq \theta$ (resp. $\overline{\nabla H}(\mathcal{I}) \neq \theta$), then $\overline{H}(\mathcal{I}) \bowtie \theta$ (resp. $\overline{\nabla H}(\mathcal{I}) \bowtie \theta$) is decidable for $\bowtie \in \{\leq, <, >, \geq\}$.*

6 Conclusion

We have studied the complexity of computing (maximum) entropy/entropy rate of Markovian models including MCs, IMCs and MDPs. We obtained a characterisation of entropy rate for general MCs based on which the entropy approximation problem and threshold problem can be solved efficiently assuming number-theoretic conjectures. For IMCs/MDPs, we obtained

polynomial-time algorithms to approximate the maximum entropy/entropy rate via convex programming, which improved a result in [7]. We also obtained conditional decidability for the threshold problem.

Open problems include unconditional polynomial-time algorithms for the entropy threshold problem for MCs and unconditional decidability for maximum entropy threshold problem for IMCs/MDPs. Furthermore, we believe it would be promising to explore more algorithmic aspects of information theory along the line of the current work, for instance, for timed automata [2].

References

1 Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.

2 Nicolas Basset. A maximal entropy stochastic process for a timed automaton,. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *ICALP (2)*, volume 7966 of *Lecture Notes in Computer Science*, pages 61–73. Springer, 2013.

3 Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Society for Industrial and Applied Mathematics, 1987.

4 Michael Benedikt, Rastislav Lenhardt, and James Worrell. Ltl model checking of interval markov chains. In Nir Piterman and Scott A. Smolka, editors, *TACAS*, volume 7795 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2013.

5 Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2011.

6 Fabrizio Biondi, Axel Legay, Pasquale Malacaria, and Andrzej Wasowski. Quantifying information leakage of randomized protocols. In Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors, *VMCAI*, volume 7737 of *Lecture Notes in Computer Science*, pages 68–87. Springer, 2013.

7 Fabrizio Biondi, Axel Legay, Bo Friis Nielsen, and Andrzej Wasowski. Maximizing entropy over markov processes. In Adrian Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, *LATA*, volume 7810 of *Lecture Notes in Computer Science*, pages 128–140. Springer, 2013.

8 Fabrizio Biondi, Axel Legay, Louis-Marie Traonouez, and Andrzej Wasowski. Quail: A quantitative security analyzer for imperative code. In Sharygina and Veith [28], pages 702–707.

9 Michele Boreale. Quantifying information leakage in process calculi. *Inf. Comput.*, 207(6):699–725, 2009.

10 Richard P. Brent. Fast multiple-precision evaluation of elementary functions. *J. ACM*, 23(2):242–251, 1976.

11 Pavol Cerný, Krishnendu Chatterjee, and Thomas A. Henzinger. The complexity of quantitative information flow problems. In *CSF*, pages 205–217. IEEE Computer Society, 2011.

12 Rohit Chadha and Michael Ummels. The complexity of quantitative information flow in recursive programs. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *FSTTCS*, volume 18 of *LIPIcs*, pages 534–545. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012.

13 Krishnendu Chatterjee, Koushik Sen, and Thomas A. Henzinger. Model-checking omega-regular properties of interval Markov chains. In Roberto M. Amadio, editor, *FoSSaCS*, volume 4962 of *Lecture Notes in Computer Science*, pages 302–317. Springer, 2008.

14 Taolue Chen and Tingting Han. On the complexity of computing maximum entropy for Markovian models. Technical report, Middlesex University London, 2014. Available via `http://www.cs.mdx.ac.uk/staffpages/taoluechen/pub-papers/fsttcs14-full.pdf`.

15 Taolue Chen, Tingting Han, and Marta Z. Kwiatkowska. On the complexity of model checking interval-valued discrete time markov chains. *Inf. Process. Lett.*, 113(7):210–216, 2013.

16 Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., New York, NY, USA, 1991.

17 Luca de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In Jos C. M. Baeten and Sjouke Mauw, editors, *CONCUR*, volume 1664 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1999.

18 Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. A note on the complexity of comparing succinctly represented integers, with an application to maximum probability parsing. *TOCT*, 6(2):9, 2014.

19 Valerie Girardin. Entropy maximization for Markov and semi-Markov processes. *Methodology and Computing in Appplied Probability*, 6:109–127, 2004.

20 Martin Grotschel, Laszlo Lovasz, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Algorithms and Combinatorics. Springer-Verlag, 1987.

21 John G. Kemeny and J. Snell. *Finite Markov Chains*. Undergraduate Texts in Mathematics. Springer-Verlag, 3rd printing 1983 edition edition, 1983.

22 Igor Kozine and Lev V. Utkin. Interval-valued finite Markov chains. *Reliable Computing*, 8(2):97–113, 2002.

23 A. J. Macintyre and A. J. Wilkie. On the decidability of the real exponential field. *Odifreddi, P. G., Kreisel 70th Birthday Volume, CLSI*, 1995.

24 William Parry. Intrinsic markov chains. *Trans. Amer. Math. Soc.*, 112:55–66, 1964.

25 Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Polynomial-time verification of pctl properties of mdps with convex uncertainties. In Sharygina and Veith [28], pages 527–542.

26 Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

27 Koushik Sen, Mahesh Viswanathan, and Gul Agha. Model-checking Markov chains in the presence of uncertainties. In Holger Hermanns and Jens Palsberg, editors, *TACAS*, volume 3920 of *Lecture Notes in Computer Science*, pages 394–410. Springer, 2006.

28 Natasha Sharygina and Helmut Veith, editors. *Computer Aided Verification – 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*. Springer, 2013.

29 Alex J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential functions. *J. Amer. Math. Soc.*, 9:1051–1094, 1996.

30 Hirotoshi Yasuoka and Tachio Terauchi. Quantitative information flow – verification hardness and possibilities. In *CSF*, pages 15–27. IEEE Computer Society, 2010.