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

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 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 [13, 6, 29]. 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.

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]

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

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, [29, 13] 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. [4] studied entropy in process algebra. These models and questions are considerably different from ours. [14, 27, 15, 25, 3] 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 [12] 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* $[\mathcal{I}] \neq \varnothing$, 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 can be easily generalised to *joint entropy*, which is 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 [12]). 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 [13] 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 [9, 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

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

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 [14].

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:

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

(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 $\vec{y}$:

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

▶ 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. [12].

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

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

where $\widetilde{\Gamma}(\vec{\lambda}_s, \nu_s, \vec{y}) = -\vec{b}_s^{\mathrm{T}} \vec{\lambda}^u_s + \vec{a}_s^{\mathrm{T}} \vec{\lambda}^l_s - \nu_s + e^{-1} \log e \cdot 2^{\nu_s} \cdot (\sum_{t \in S} 2^{\vec{\lambda}^u_{s,t} - \vec{\lambda}^l_{s,t} + y_t})$ 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 [10, 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. [12]).

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 [12].

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 [12].

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 [28]. 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 the 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 E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.

2 N. Basset. A maximal entropy stochastic process for a timed automaton,. In *ICALP (2)*, LNCS 7966, pages 61–73. Springer, 2013.

3 M. Benedikt, R. Lenhardt, and J.B. Worrell. LTL model checking of interval Markov chains. In *TACAS*, LNCS 7795, pages 32–46. Springer, 2013.

4 M. Boreale. Quantifying information leakage in process calculi. *Inf. Comput.* 207(6): 699-725, 2009.

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

6 F. Biondi, A. Legay, P. Malacaria, and A. Wasowski. Quantifying information leakage of randomized protocols. In *VMCAI*, LNCS 7737, pages 68–87. Springer, 2013.

7 F. Biondi, A. Legay, B.F. Nielsen, and A. Wasowski. Maximizing entropy over Markov processes. In *LATA*, LNCS 7810, pages 128–140. Springer, 2013.

8 F. Biondi, A. Legay, L.-M Traonouez, and A. Wasowski. Quail: A quantitative security analyzer for imperative code. In *CAV*, LNCS 8044, pages 702–707. Springer, 2013.

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

10 A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 1987.

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

12 T. Chen and T. Han. On the Complexity of Computing Maximum Entropy for Markovian Models. Technical report, available via `www.cs.mdx.ac.uk/staffpages/taoluechen/pub-papers/fsttcs14-full.pdf`, 2014.

13 R. Chadha and M. Ummels. The complexity of quantitative information flow in recursive programs. In *FSTTCS*, volume 18 of *LIPIcs*, pages 534–545.

14 K. Chatterjee, K. Sen, and T. A. Henzinger. Model-checking omega-regular properties of interval Markov chains. In *FoSSaCS*, LNCS 4962, pages 302–317. Springer, 2008.

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

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

17 L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *CONCUR*, LNCS 1664, pages 66–81. Springer, 1999.

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

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

20 M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1987.

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

22 I. Kozine and L.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 W. Parry. Intrinsic markov chains. *Trans. Amer. Math. Soc.*, 112:55–66, 1964.

25 A. Puggelli, W. Li, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *CAV*, LNCS 8044, pages 527–542. Springer, 2013

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

27 K. Sen, M. Viswanathan, and G. Agha. Model-checking Markov chains in the presence of uncertainties. In *TACAS*, volume LNCS 3920, pages 394–410. Springer, 2006.

28 A. 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.

29 H. Yasuoka and T. Terauchi. Quantitative information flow - verification hardness and possibilities. In *CSF*, pages 15–27. IEEE Computer Society, 2010.

A Proof of Theorem 8 and Theorem 10

Proof. It follows from the definition of $\mathbb{H}(Y|X)$ that

$$\begin{aligned}\mathbb{H}(X_n \mid X_{n-1}) &= \sum_{s,t\in S} \Pr(X_n = t, X_{n-1} = s) \log \frac{\Pr(X_{n-1} = s)}{\Pr(X_n = t, X_{n-1} = s)} \\ &= -\sum_{s,t\in S} \Pr(X_n = t, X_{n-1} = s) \log \Pr(X_n = t|X_{n-1} = s) \\ &= -\sum_{s\in S} \Pr(X_{n-1} = s) \sum_{t\in S} \Pr(X_n = t|X_{n-1} = s) \log \Pr(X_n = t|X_{n-1} = s) \\ &= \sum_{s\in S} \Pr(X_{n-1} = s) L(s)\end{aligned}$$

and thus

$$\begin{aligned}\mathbb{H}(X_0, \cdots, X_n) &= \sum_{i=1}^{n} \mathbb{H}(X_i|X_0, \cdots, X_{i-1}) + \mathbb{H}(X_0) \\ &= \sum_{i=1}^{n} \mathbb{H}(X_i|X_{i-1}) + \mathbb{H}(X_0) \\ &= \sum_{i=1}^{n} \left(\sum_{s\in S} \Pr(X_{i-1} = s) L(s) \right) + \mathbb{H}(X_0) \\ &= \sum_{s\in S} L(s) \sum_{i=0}^{n-1} \Pr(X_i = s) + \mathbb{H}(X_0)\end{aligned}$$

It follows that

$$\begin{aligned}H(\mathcal{D}) = \mathbb{H}(X_0, X_1, \cdots) &= \sum_{s\in S} L(s) \sum_{i=0}^{\infty} \Pr(X_i = s) + \mathbb{H}(X_0) \\ &= \sum_{s\in S} L(s) \sum_{i=0}^{\infty} \pi_s^{(i)} + \mathbb{H}(\alpha) = \sum_{s\in S} L(s)\xi(s) + \mathbb{H}(\alpha),\end{aligned}$$

and

$$\begin{aligned}\nabla H(\mathcal{D}) &= \lim_{n\to\infty} \frac{1}{n} \mathbb{H}(X_0, \cdots, X_n) \\ &= \lim_{n\to\infty} \frac{1}{n} \left(\sum_{s\in S} L(s) \sum_{i=0}^{n-1} \Pr(X_i = s) + \mathbb{H}(X_0) \right) \\ &= \sum_{s\in S} L(s) \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \Pr(X_i = s) \\ &= \sum_{s\in S} L(s) \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \pi_s^{(i)} \\ &= \sum_{s\in S} L(s)\zeta(s).\end{aligned}$$

This completes the proof for Theorem 8 and Theorem 10. ◀

B Proofs for Section 4

We start from some general introduction of complexity theory, regarding *straight-line programs* and the *counting hierarchy*; see [1] for details.

We assume a countable set of variables ranged over by $x, y, z, \cdots$. A (division-free) straight-line program is a finite list of instructions of the form $x \leftarrow c$ or $x \leftarrow y * z$, where $c \in \{0, 1\}$, $* \in \{+, -, \cdot\}$. Such a program is closed if all variables that appear on the right-hand side of an instruction also appear on the left-hand side of a preceding instruction. Clearly a closed straight-line program represents an integer which is the value of the last variable that is assigned to. The problem *PosSLP* is to decide, given a closed straight-line program, whether the corresponding integer is *positive*.

The counting hierarchy consists of the classes CH_i where $\mathsf{CH}_0 = \mathsf{P}$ and $\mathsf{CH}_{i+1} = \mathsf{PP}^{\mathsf{CH}_i}$ for all $i \in \mathbb{N}$. Here PP refers to probabilistic polynomial time, i.e. the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of less than $\frac{1}{2}$ for all instances. Allender *et al.* [1] showed that PosSLP belongs to the complexity class $\mathsf{P}^{\mathsf{CH}_3}$ and thus to the fourth-level of the counting hierarchy. We note that counting hierarchy is contained in PSPACE, but it unlikely to be complete to PSPACE.

B.1 Proof of Proposition 11

Proof. We write, for each i, $p_i = \frac{m_i}{n_i}$ and $q_i = \frac{m_i'}{n_i'}$ where $m_i, n_i, m_i', n_i' \in \mathbb{N}$. Let $N = \gcd\{n_i \mid 1 \leq i \leq n\}$. Note that the size of N is polynomial in the size of p_i's as $N \leq \prod_{1 \leq i \leq n} p_i$. Furthermore $\sum_{i=1}^{n} p_i \log q_i = \frac{1}{N} \sum_{i=1}^{n} m_i (\frac{N}{n_i})(\log m_i' - \log n_i')$. The conclusion follows from the same argument of [13, Lemma 4]. ◀

B.2 Proof of Proposition 12

Proof. We write, for each i, $p_i = \frac{m_i}{n_i}$ and $q_i = \frac{m_i'}{n_i'}$ where $m_i, n_i, m_i', n_i' \in \mathbb{N}$ and $\theta = \frac{\theta_1}{\theta_2}$ where $\theta_1, \theta_2 \in \mathbb{N}$. Let $N = \gcd\{n_i \mid 1 \leq i \leq n, \theta_2\}$. Note that the size of N is polynomial in the size of p_i's and θ_2 as $N \leq \prod_{1 \leq i \leq n} p_i \cdot \theta_2$. Evidently, $\sum_{i=1}^{n} p_i \log q_i = \theta$ can be rearranged into the form of equality of *product of exponentials* and the conclusion follows from [18, Propoositon 2.1]. ◀

B.3 Proof of Proposition 16

This result is a "folklore" result in MC theory. However, we are not aware of a documented proof. It can be derived from the ergodic theorem without much difficulty. Instead, here we provide a purely algebraic proof which might be of independent interests.

Proof. Recall that by definition, $\zeta(s) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \pi_s^{(i)} = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} (\alpha \mathbf{P}^i)_s$. Namely,

$$\vec{\zeta} = \alpha \cdot \lim_{n \to \infty} \frac{1}{n} (\mathbf{I} + \mathbf{P} + \cdots + \mathbf{P}^{n-1})$$

For general MCs, the transition probability matrix $\mathbf{P}$ has the form

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

As $\mathbf{P}$ is a stochastic matrix, the *Cesaro sum* $\lim_{n\to\infty}\frac{1}{n}(\mathbf{I}+\mathbf{P}+\cdots+\mathbf{P}^{n-1})$ exists and must be the case that

$$\lim_{n\to\infty}\frac{1}{n}(\mathbf{I}+\mathbf{P}+\cdots+\mathbf{P}^{n-1})=G,$$

where G satisfies

$$(\mathbf{I}-\mathbf{P})G=0.$$

Firstly note that $\mathbf{P}$ is an upper-triangular(block) matrix, hence so is G. Together with $G=\mathbf{P}G$, we can further write

$$G=\begin{pmatrix} G_{00} & G_{01} & G_{02} & \cdots & G_{0h} \\ \mathbf{0} & G_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & G_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & G_h \end{pmatrix}$$

Consider $B_iG_i=\mathbf{0}$ for $1\le i\le h$. Note that G_i is an irreducible nonnegative matrix, by the Perron–Frobenius theorem, or by the fundamental matrix in MCs ([21]), we have that $G_i=\mathbf{1}^{\mathrm{T}}\vec{y}_i$ where $\vec{y}_i$ is the solution of the following system of linear equations:

$$\vec{y}_iB_i=\vec{y}_i \text{ and } \mathbf{1}^{\mathrm{T}}\vec{y}=1$$

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

Furthermore, $QG_{00}=G_{00}$ and for each $1\le i\le h$,

$$QG_{0i}+R_iG_i=G_{0i}.$$

As $\mathbf{I}-Q$ is invertible [21], we have that $G_{00}=\mathbf{0}$, and for each $1\le i\le h$,

$$G_{0i}=(\mathbf{I}-Q)^{-1}R_iG_i=(\mathbf{I}-Q)^{-1}R_i\mathbf{1}^{\mathrm{T}}\vec{y}_i.$$

Concluding,

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

We are done.

◀

C Proofs for Section 5

C.1 Proof of Proposition 20

As remarked before, (3) is the analogy of the linear program for the expected total reward objective in MDPs. The proof also follows this line.

Proof. We start by considering the following Bellman equation:

$$y_s = \begin{cases} \max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y}) & s \notin G \\ 0 & s \in G \end{cases} \tag{7}$$

(7) can be written as a functional $\mathcal{F} : [0,1]^{|S|} \to [0,1]^{|S|}$ such that $[\mathcal{F}(\vec{y})]_s = \max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y})$ if $s \notin G$ and 0 otherwise. One can easily verify that $\mathcal{F}$ is monotonic over the complete lattice $([0,1]^{|S|}, \leq)$, and hence $\mathcal{F}$ admits a least fixpoint $\mathsf{lfx}(\mathcal{F})$. We shall prove that the $\mathsf{lfx}(\mathcal{F})$ captures $\overline{H}(\mathcal{I}) - \mathbb{H}(\alpha)$, namely $\alpha \cdot \mathsf{lfx}(\mathcal{F}) = \overline{H}(\mathcal{I}) - \mathbb{H}(\alpha)$.

On the one hand, for *any* fixpoint of $\mathcal{F}$, say $\vec{y}$, we can obtain vectors $\vec{x}_s = (x_{st})_{t \in S}$ for each state $s \in S \setminus G$ such that $\vec{x}_s = \arg\max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y})$ and thus $\vec{y}_s = \Gamma(\vec{x}_s, \vec{y})$. Clearly by the definition of the constraints of $\Omega(s)$, we can construct an MC $\mathcal{D}$ where for $s \in S \setminus G$, transition probabilities are entries $\vec{x}_s$ and for $s \in G$, it is absorbing. Clearly $\mathcal{D} \in [\mathcal{I}]$. Furthermore, by Proposition 14 and Theorem 8,

$$\sum_{s \in S} \alpha(s) \vec{y}_s = \sum_{s \in S \setminus G} \alpha(s) \vec{y}_s = H(\mathcal{D}) - \mathbb{H}(\alpha).$$

It follows that

$$\alpha \cdot \mathsf{lfx}(\mathcal{F}) \leq \overline{H}(\mathcal{I}) - \mathbb{H}(\alpha).$$

On the other hand, for any MC $\mathcal{D} \in [\mathcal{I}]$, we can take its transition probabilities to form $\vec{x}_s$ for each $s \in S \setminus G$ (note that states in G are absorbing). By definition of (7), it must be the case that

$$[\mathsf{lfx}(\mathcal{F})]_s \geq \Gamma(\vec{x}_s, \mathsf{lfx}(\mathcal{F}))$$

for $s \in S \setminus G$. It follows from Proposition 14 and Theorem 8 that

$$\alpha \cdot [\mathsf{lfx}(\mathcal{F})] \geq H(\mathcal{D}) - \mathbb{H}(\alpha),$$

which implies that

$$\alpha \cdot [\mathsf{lfx}(\mathcal{F})] \geq \overline{H}(\mathcal{I}) - \mathbb{H}(\alpha).$$

Concluding,

$$\alpha \cdot [\mathsf{lfx}(\mathcal{F})] = \overline{H}(\mathcal{I}) - \mathbb{H}(\alpha).$$

By a standard argument [26, 5], the least solution of $\mathcal{F}$ in terms of (7) can be computed by solving the following non-linear program

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

which is exactly (3). ◀

C.2 Solving (3) from Section 5.1

Recall the non-linear program (3).

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

In this section, we give details on how to solve it, elaborating the sketch in the main text. Observe that here we have a nested optimisation problem because of the presence of an inner optimisation $\max_{\vec{x}_s\in\Omega(s)} \Gamma(\vec{x}_s, \vec{y})$ which we shall handle firstly. For simplicity, we rewrite it in a more explicit form as

$$\text{maximise} \quad \sum_{i=1}^{n} c_i x_i - \sum_{i=1}^{n} x_i \log x_i \tag{8}$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_i = 1 \tag{9}$$

$$a_i \leq x_i \leq b_i \qquad \forall i. 1\leq i\leq n \tag{10}$$

Here $n = |S|$, $\vec{c} = (c_i)_{i=1}^n = (y_t)_{t\in S} \geq 0$, $\vec{a} = (a_i)_{i=1}^n = (\mathbf{P}^l(s,t))_{t\in S}$ and $\vec{b} = (b_i)_{i=1}^n = (\mathbf{P}^u(s,t))_{t\in S}$.

Observe that the objective function (8) is a *concave* function and is to be *maximised*, and all the constraints are linear. Hence (8)-(10) is a convex program [2]. The main strategy is to apply the Lagrange duality. To this end, for each $1\leq i\leq n$, we introduce a single variable ν and two vectors of variables $\vec{\lambda}^u$ and $\vec{\lambda}^l$ corresponding to (9) and (10), respectively. For simplicity we write $\vec{\lambda}$ as the concatenation of $\vec{\lambda}^l$ and $\vec{\lambda}^u$.

We now aim to derive the Lagrange dual function. For this purpose, we consider the *conjugate function*. Let $f:\mathbb{R}^n\to\mathbb{R}$. In a nutshell, the function $f^\star:\mathbb{R}^n\to\mathbb{R}$, defined as

$$f^\star(\vec{y}) = \sup_{\vec{x}} (\vec{y}^T\vec{x} - f(\vec{x}))$$

is called the *conjugate* of the function f.

▶ **Lemma 26.** *For the function*

$$f(\vec{x}) = \sum_{i=1}^{n} c_i x_i - \sum_{i=1}^{n} x_i \log x_i$$

with domain $\mathbb{R}^n_{\geq 0}$*, the conjugate function is*

$$f^\star(\vec{y}) = -e^{-1}\log e \sum_{i=1}^{n} 2^{-y_i+c_i} \quad .$$

Proof. By definition, we compute the conjugate as

$$f^\star(\vec{y}) = \sup_{\vec{x}} \{\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} (c_i - \log x_i) x_i\}.$$

By setting the partial derivatives wrt. x_i $(1 \leq i \leq n)$ to be 0, we have that for each i, $y_i - c_i + \log x_i + \log e = 0$, i.e., $\log x_i = -y_i - \log e + c_i$ and thus $x_i = e^{-1}2^{-y_i+c_i}$. It follows that

$$f^\star(\vec{y}) = -e^{-1}\log e \sum_{i=1}^{n} 2^{-y_i+c_i}$$

(Note that we are of base 2.) ◀

It is known that [2, Chapter 5] the conjugate function and Lagrange dual function are closely related. Indeed, consider an optimisation problem with linear inequality and equality constraints,

$$\begin{aligned} \text{minimise} \quad & h(x) \\ \text{subject to} \quad & F\vec{x} \preceq \vec{f} \\ & G\vec{x} = \vec{g}. \end{aligned}$$

We have that the *Lagrangian dual function* as

$$\tilde{h}(\lambda, \nu) = -\vec{f}^{\mathrm{T}}\lambda - \vec{g}^{\mathrm{T}}\nu - h^\star(-F^{\mathrm{T}}\lambda - G^{\mathrm{T}}\nu).$$

Applying this result to Γ with $F = (\mathbf{I} \vdots -\mathbf{I})^{\mathrm{T}}$, $\vec{f} = (\vec{b}, -\vec{a})^{\mathrm{T}}$, $G = (1, \cdots, 1)^{\mathrm{T}}$ and $\vec{g} = 1$ (note that F is of $2n \times n$, $\vec{f}$ is of $2n \times 1$, G is of $n \times 1$ and $\vec{g}$ is of 1×1), we obtain

$$\widetilde{\Gamma}(\vec{\lambda}, \nu, \vec{c}) = -\vec{f}^{\mathrm{T}}\lambda - \nu + e^{-1}\log e \sum_{i=1}^{n} 2^{f_i^{\mathrm{T}}\lambda+\nu+c_i} = -f^{\mathrm{T}}\lambda - \nu + e^{-1}\log e \cdot 2^{\nu} \cdot \sum_{i=1}^{n} 2^{f_i^{T}\lambda+c_i}$$

where f_i is the i-th column of F. To simplify further we obtain

$$\widetilde{\Gamma}(\vec{\lambda}, \nu, \vec{c}) = -\vec{b}^{\mathrm{T}}\vec{\lambda}^u + \vec{a}^{\mathrm{T}}\vec{\lambda}^l - \nu + e^{-1}\log e 2^{\nu} \sum_{i=1}^{n} 2^{\vec{\lambda}_i^u - \vec{\lambda}_i^l + c_i}$$

under $\vec{\lambda} \geq 0$ and $\nu \geq 0$.

The Slater's condition [2] is a sufficient condition for strong duality to hold for a convex optimization problem, which states, informally, that the feasible region must have an interior point. In our case, the Slater condition is easy to verify, as all the constraints are linear. Hence we have the strong duality:

▶ Proposition 27.

$$\max_{\vec{x} \in \Omega(s)} \Gamma(\vec{x}, \vec{y}) = \min_{\vec{\lambda} \geq 0, \nu \geq 0} \widetilde{\Gamma}(\vec{\lambda}, \nu, \vec{y})$$

It follows that (3) can be rewritten as a non-linear program over $\vec{y}$ and $\vec{\lambda}_s^l$ which is the concatenation of $\vec{\lambda}_s^l = (\lambda_{s,t}^l)_{t \in S}$ and $\vec{\lambda}_s^u = (\lambda_{s,t}^u)_{t \in S}$, and ν_s for each $s \in S \setminus G$.

$$\begin{aligned} \text{minimise} \quad & \sum_{s \in S \setminus G} \alpha(s) y_s \\ \text{subject to} \quad & y_s \geq \min_{\vec{\lambda}_s, \nu_s} \widetilde{\Gamma}(\vec{\lambda}_s, \nu_s, \vec{y}) && s \notin G \\ & y_s = 0 && s \in G \\ & \lambda_{s,t}^l \geq 0, \lambda_{s,t}^u, \nu_s \geq 0 && s \notin G, t \in S \end{aligned}$$

which is equivalent to (4)

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

It remains to check (4) is a convex program. For this purpose, it suffices to check the constraints are jointly convex. This is indeed the case:

▶ Proposition 28. (4) is a convex program.

Proof. It suffices to show that

$$\widetilde{\Gamma}(\vec{\lambda}, \nu, \vec{y}) = -\vec{b}^{\mathrm{T}}\vec{\lambda}^u + \vec{a}^{\mathrm{T}}\vec{\lambda}^l - \nu + e^{-1}\log e \cdot 2^{\nu} \cdot \sum_{i=1}^{n} 2^{\vec{\lambda}^u_i - \vec{\lambda}^l_i + \vec{y}_i}$$

in the constraint of (4) is convex. Note that linear and exponential functions are convex, and convexity is invariant under affine maps and is closed under addition [2]. The conclusion follows. ◀

The following proposition is a standard result, see e.g. [10].

▶ Proposition 29. Given the convex program

$$\begin{aligned} \text{minimise} \quad & f(\vec{x}) \\ \text{subject to} \quad & f_i(\vec{x}) \leq 0 \end{aligned} \tag{11}$$

with $\vec{x} \in \mathbb{R}^n$ and f_i are convex functions. The optimum can be found within $\epsilon > 0$ in time complexity that is polynomial in the size of the problem and $\log(\frac{1}{\epsilon})$.

The main result, Theorem 21 hence follows.

C.3 Finiteness of maximum entropy rate

▶ Proposition 30. For any IMC $\mathcal{I}$, $\overline{\nabla H}(\mathcal{I})$ is finite.

Proof. For each $\mathcal{D} \in [\mathcal{I}]$, note that $L(s) \leq |S| \log |S|$, hence by Theorem 10, $\nabla H(\mathcal{D}) \leq |S| \log |S|$. Note that $\sum_{s\in S} \zeta(s) = 1$. It follows that $\overline{\nabla H}(\mathcal{I}) \leq |S| \log |S|$. ◀

C.4 Proof of Proposition 22

As remarked before, (6) and (5) are the analogy of the linear program for the limiting average reward objective in MDPs. The proof also follows this line.

Proof. We first show that (5) captures the maximum entropy rate for an SCC $B \in \mathcal{E}(\mathcal{I})$, i.e., $\overline{\nabla H}(B)$. This corresponds to the "unichain MDP" (cf. [26, Chapter 8]). There are two different ways to view this problem leading to different proofs. For completeness we give both sketches. Similar to Proposition 20, we consider the following Bellman equation:

$$y_s + r = \max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y}) \tag{12}$$

which can rewrite as a functional $\mathcal{F}$. Following a similar argument as in Proposition 20, one can show the least fixpoint of $\mathcal{F}$, $\mathsf{lfx}(\mathcal{F})$ captures $\overline{\nabla H}(B)$.

Alternatively, we consider the following non-linear program:

$$\begin{aligned}
\text{maximise} \quad & \sum_{s\in S} y_s(-\sum_{t\in S} x_{s,t}\log x_{s,t}) \\
\text{subject to} \quad & y_s = \sum_{t\in S} x_{s,t}y_t \quad \forall s \in B \\
& \sum_{s\in B} y_s = 1 \\
& \mathbf{P}^l(s,t) \le x_{s,t} \le \mathbf{P}^u(s,t) \quad s,t \in B \\
& \sum_{t\in S} x_{s,t} = 1 \quad s \in B
\end{aligned} \tag{13}$$

It is rather straightforward to verify that the optimal solution of (13) captures $\overline{\nabla H}(\mathcal{I})$ by Proposition 16 and Theorem 10, noting that intuitively $\vec{y} = (y_s)_{s\in S}$ encodes the stationary distribution and $(-\sum_{t\in B} x_{s,t}\log x_{s,t})$ in the objective function is the local entropy of s. Now, simply observe that (5) and (13) are the primal and the dual problem. The conclusion follows.

We then turn to (6), which is for the general IMC (instead of an SCC). As in Proposition 20, we consider the Bellman equation

$$z_s = \max\{\max_{\vec{x}_s\in\Omega(s)} \Lambda(\vec{x}_s,\vec{z}), r_B\} \tag{14}$$

which can be written as a functional $\mathcal{F} : [0,1]^{|S|} \to [0,1]^{|S|}$ such that

$$[\mathcal{F}(\vec{z})]_s = \max\{\max_{\vec{x}_s\in\Omega(s)} \Lambda(\vec{x}_s,\vec{z}), r_B\}.$$

Recall that for each s there is a unique SCC B to which s belongs. One can easily verify that $\mathcal{F}$ is monotonic over the complete lattice $([0,1]^{|S|}, \le)$, and hence $\mathcal{F}$ admits a least fixpoint $\mathsf{lfx}(\mathcal{F})$. We shall prove that the $\mathsf{lfx}(\mathcal{F})$ captures $\overline{\nabla H}(\mathcal{I})$, namely $\alpha\cdot\mathsf{lfx}(\mathcal{F}) = \overline{\nabla H}(\mathcal{I})$.

On the one hand, for any fixpoint of $\mathcal{F}$, say $\vec{z}$, we have two cases:

- $z_s = \max_{\vec{x}_s\in\Omega(s)} \Lambda(\vec{x}_s,\vec{z})$. In this case we obtain vectors $\vec{x}_s$ such that $\vec{z}_s = \Lambda(\vec{x}_s,\vec{z})$.
- $z_s = r_B$. In this case, we resort to (5) or equally (13), and obtain vectors $\vec{x}_s$ from their solution as well.

Clearly by the definition of the constraints in $\Omega(s)$, we can construct an MC $\mathcal{D}$ consisting of entries $\vec{x}_s$ as its transition probabilities such that $\mathcal{D} \in [\mathcal{I}]$. Furthermore, by Proposition 16 and Theorem 10 $\sum_{s\in S}\alpha(s)\vec{z}_s = \nabla H(\mathcal{D})$. It follows that

$$\alpha\cdot\mathsf{lfx}(\mathcal{F}) \le \overline{\nabla H}(\mathcal{I}).$$

On the other hand, for *any* MC $\mathcal{D} \in [\mathcal{I}]$, we can take its transition probabilities to form $\vec{x}_s$ for each $s \in S$. Furthermore, by definition of (14), it must be the case that

$$[\mathsf{lfx}(\mathcal{F})]_s \ge \Lambda(\vec{x}_s, \mathsf{lfx}(\mathcal{F})) \text{ and } [\mathsf{lfx}(\mathcal{F})]_s \ge r_B.$$

It follows from Proposition 16 and Theorem 10 that $\alpha\cdot[\mathsf{lfx}(\mathcal{F})] \ge \nabla H(\mathcal{D})$ which implies that $\alpha\cdot[\mathsf{lfx}(\mathcal{F})] \ge \overline{\nabla H}(\mathcal{I})$.

By a standard argument [5], the least solution of $\mathcal{F}$ in terms of (14) can be computed by the non-linear program (6). This completes the proof. ◀

C.5 Proof of Proposition 23

Proof. Observe that $g(\vec{r})$ can be write as $\mathbf{P} \cdot \vec{r}$ for some $\mathbf{P}$ consisting of the optimum $\vec{x}_s$ as entries. Recall that $\|\cdot\|$ denotes the ∞-norm. It follows that

$$\|\max_{\mathbf{P}} \mathbf{P}\vec{r} - \max_{\mathbf{P}'} \mathbf{P}'\vec{r}'\| \le \|\mathbf{P}\vec{r} - \mathbf{P}\vec{r}'\| = \|\mathbf{P}\| \cdot \|\vec{r} - \vec{r}'\| \le \|\mathbf{P}\| \cdot \epsilon \le \epsilon.$$

Hence

$$|g(\vec{r}) - g(\vec{r}')| = |\alpha \cdot \max_{\mathbf{P}} \mathbf{P}\vec{r} - \alpha \cdot \max_{\mathbf{P}'} \mathbf{P}'\vec{r}'| \le \epsilon.$$

◀

C.6 Threshold problem

Entropy. We start to give more details for the entropy. By Theorem 8 and Proposition 14, it is easy to see that $\overline{H}(\mathcal{I})$ is equal to the optimal solution of

$$\begin{aligned} \text{maximise} \quad & \sum_{s \in S \setminus G} \alpha(s) \vec{y}_s \\ \text{subject to} \quad & y_s = \sum_{t \in S} x_{s,t} y_t - \sum_{t \in S} x_{s,t} \log x_{s,t} \quad \forall s \in S \setminus G \\ & y_s = 0 \quad \forall s \in G \\ & \mathbf{P}^l(s,t) \le x_{s,t} \le \mathbf{P}^u(s,t) \\ & \sum_{t \in S} x_{s,t} = 1 \end{aligned} \tag{15}$$

▶ Remark. It can be shown that (15) gives an alternative way to compute $\overline{H}(\mathcal{I})$ which is actually used in [7] and is arguably more elegant. However, it *per se* is *not* a convex program (for instance, note the bilinear form $\sum_{t \in S} x_{s,t} y_t$), so does *not* lead to polynomial-time bound for the approximation problem outright. Nevertheless they are sufficient to derive the results in this section.

(15) can be encoded into the first-order theory. Indeed deciding $\overline{H}(\mathcal{I}) \ge \theta$ amounts to checking

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

where $\vec{x}$ is the concatenation of $\vec{x}_s = (x_{s,t})_{t \in S}$ for $s \in S \setminus G$ and $\vec{y} = (y_s)_{s \in S}$. We are done.

Entropy rate. For the entropy rate, we take (5) which we expand as

$$\begin{aligned} \text{minimise} \quad & r_B \\ \text{subject to} \quad & r_B + y_s = \sum_{t \in S} x_{s,t} y_t - \sum_{t \in S} x_{s,t} \log x_{s,t} \\ & \mathbf{P}^l(s,t) \le x_{s,t} \le \mathbf{P}^u(s,t) \\ & \sum_{t \in S} x_{s,t} = 1 \end{aligned}$$

together with (6) which we expand as

$$\begin{aligned}
\text{minimise} \quad & \sum_{s\in S\setminus G} \alpha(s)\vec{z}_s \\
\text{subject to} \quad & z_s \geq \sum_{t\in S} x_{s,t} z_t \quad \forall s \in S \\
& z_s \geq r_B \quad \forall s \in G \\
& \mathbf{P}^l(s,t) \leq x_{s,t} \leq \mathbf{P}^u(s,t) \\
& \sum_{t\in S} x_{s,t} = 1
\end{aligned}$$

Deciding $\overline{\nabla H}(\mathcal{I}) \geq \theta$ amounts to checking

$$\forall \vec{z} \exists \vec{x}, \vec{y}, \vec{r}. \bigwedge \begin{cases} \sum_{s\in S} \alpha(s) \cdot \vec{z}_s \geq \theta \\ z_s \geq \sum_{t\in S} x_{s,t} z_t \quad \forall s \in S \\ z_s \geq r_B \quad \forall B \in \mathcal{E}(\mathcal{I}) \\ r_B + z_s = \sum_{t\in S} x_{s,t} z_t - \sum_{t\in S} x_{s,t} \log x_{s,t} \quad \forall B \in \mathcal{E}(\mathcal{I}) \wedge s \in B \\ \mathbf{P}^l(s,t) \leq x_{s,t} \leq \mathbf{P}^u(s,t) \quad \forall s \in S \\ \sum_{t\in S} x_{s,t} = 1 \quad \forall s \in S \end{cases}$$

where $\vec{x}$ is the concatenation of $\vec{x}_s = (x_{s,t})_{t\in S}$ for $s \in S$, $\vec{y} = (y_s)_{s\in S}$, $\vec{z} = (z_s)_{s\in S}$ and $\vec{r} = (r_B)_{B\in\mathcal{E}(\mathcal{I})}$.

D Computing the maximum entropy of MDPs

In this section, we turn our attention to MDPs, which substantiates the results claimed in Section 1, in particular, Table 1. We start from definitions.

▶ **Definition 31.** A *Markov Decision Process* (MDP) is a tuple $\mathcal{M} = (S, \alpha, Act, \mathfrak{A}, \tau)$, where

- S and α are defined the same as in Definition 1,
- Act is a finite set of actions;
- $\mathfrak{A} : S \to Act$ such that for each state $s \in S$, $\mathfrak{A}(s)$ is a set of actions which are *enabled* in s.
- $\tau : S \times Act \hookrightarrow \Delta(S)$ is a partial function s.t. for each s and $a \in \mathfrak{A}(s)$, $\sum_{t\in S} \tau(t|s,a) = 1$.

Without loss of generality, we assume that $\mathfrak{A}(s) \neq \varnothing$ for each $s \in S$. Intuitively at each state s of $\mathcal{M}$, an action a is chosen *nondeterministically* from the set $\mathfrak{A}(s)$. A successor state s' is then chosen according to the distribution $\tau(\cdot|s,a)$ with probability $\tau(s'|s,a)$.

A *path* π in $\mathcal{M}$ is a sequence of the form $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots$ where $s_i \in S$, $a_{i+1} \in \mathfrak{A}(s_i)$ and $\tau(s_{i+1}|s_i, a_{i+1}) > 0$ for each $i \geq 0$. A finite path is a prefix of an infinite path ending in a state. Let $Paths^\star$ be the set of finite paths. A *scheduler* $\sigma : Paths^\star \to \Delta(Act)$ maps a finite path $\rho = s_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n$ (the *history*) to a distribution over Act with the constraint that the support of $\sigma(\rho)$ is contained by $\mathfrak{A}(s_n)$. In particular, a *simple* scheduler σ chooses an *action* (instead of a distribution) only based on the current state and $\sigma(s) \in \mathfrak{A}(s)$ for each state s. A memoryless *randomised* scheduler σ prescribes, for each state s, a distribution ν over $\mathfrak{A}(s)$, which induces a distribution over S as $\sum_{a\in\mathfrak{A}(s)} \nu(a) \cdot \tau(s'|s,a)$ for each $s' \in S$. Note that we may obtain a DTMC by resolving all the nondeterminism in an MDP using a scheduler σ in a standard way (see, e.g., [1, 26]). In the sequel, we write $\mathcal{M}_\sigma$ for such an MC given an MDP $\mathcal{M}$ and a scheduler σ.

The following notions are standard which play a similar role as SCCs for MCs, cf. [3].

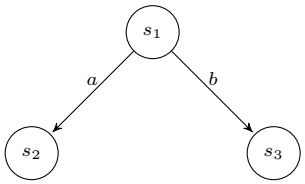

Figure 1 Example MDP1

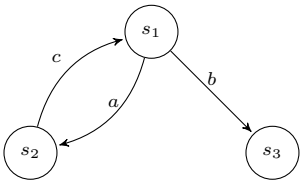

Figure 2 Example MDP2

▶ **Definition 32** (End component)**.** A pair (T, B) with $T \subseteq S$ and $B \subseteq Act$ is an *end component* (EC) of $\mathcal{M}$ if (1) for all $a \in B$, whenever $\tau(t|s,a) > 0$, $t \in T$; and (2) for all $s, t \in T$, t is reachable from s.

An EC is a *maximal end component* (MEC) if it is maximal wrt. pointwise subset ordering. We write $\mathcal{E}(\mathcal{M})$ for the set of MECs of $\mathcal{M}$.

We sometimes relax the definition of MDPs by allowing $\mathfrak{A}(s)$ to be *infinite*. In literature, this is often called *semi-infinite MDPs*. As long as $\mathfrak{A}(s)$ is compact (for instance in the paper, $\mathfrak{A}(s) \subseteq \mathbb{R}^{|S|}$ with respect to the Euclidean topology), most interesting properties for MDPs are carried over.

We are interested in maximising the entropy/entropy rate of $\mathcal{M}$ under schedulers. Note that, in general, for a given scheduler σ, $\mathcal{M}_\sigma$ as an MC is of a (countably) infinite state space [1]. However, in our setting, the associated stochastic process is still over the original state space S. So the definitions of entropy/entropy rate for $\mathcal{M}_\sigma$ apply. Nevertheless, the formal definition is given as follows:

▶ **Definition 33.** Given an IMC $\mathcal{M}$,

- the *maximum* entropy of $\mathcal{M}$, denoted by $\overline{H}(\mathcal{M})$ is defined as

$$\overline{H}(\mathcal{M}) = \sup_\sigma H(\mathcal{M}_\sigma)$$

- the *maximum* entropy rate of $\mathcal{I}$, denoted by $\overline{\nabla H}(\mathcal{M})$ is defined as

$$\overline{\nabla H}(\mathcal{M}) = \sup_\sigma \nabla H(\mathcal{M}_\sigma)$$

Our task is to compute the quantities $\overline{H}(\mathcal{M})$ and $\overline{\nabla H}(\mathcal{M})$, in the sense of the approximation problem and the threshold problem. Let's start from some observations.

▶ **Example 34.**
- Randomised schedulers are better. A simple example is depicted in Figure 1; note all the probabilistic distributions are Dirac and s_2, s_3 are absorbing states and s_1 is the initial state. For any simple scheduler σ, $H(\mathcal{M}_\sigma) = 0$. However, for the (memoryless) randomised scheduler $\sigma = [a \mapsto 0.5, b \mapsto 0.5]$, $H(\mathcal{M}_\sigma) = 1$.
- The maximum entropy might be unbounded, as in IMCs. Again a simple example is depicted in Figure 2. Suppose the scheduler $\sigma = [a \mapsto x, b \mapsto 1-x, c \mapsto 1]$. One can easily calculate that

$$H(\mathcal{M}_\sigma) = -\frac{x\log(x) + (1-x)\log(1-x)}{1-x} = -\log(1-x) - \frac{x\log(x)}{1-x}$$

The limit is ∞ when $x \to 1$, which implies that $\overline{H}(\mathcal{M}) = \infty$.

We are now in a position to tackle the approximation problem. For a given MDP $\mathcal{M} = (S, \alpha, Act, \mathfrak{A}, \tau)$, we can associate a semi-infinite MDP $\mathcal{M}' = (S, \alpha, Act', \mathfrak{A}', \tau')$ where

- $\mathfrak{A}'(s) = \{(x_t)_{t\in S} \mid x_t = \sum_{a\in\mathfrak{A}(s)} x_{s,a} \cdot \tau(t|s,a), 0 \leq x_{s,a} \leq 1, \text{ and } \sum_{a\in\mathfrak{A}(s)} x_{s,a} = 1\}$,
- $Act' = \bigcup_{s\in S} \mathfrak{A}'(s)$, and
- $\tau'(t|s,(x_t)_{t\in S}) = x_t$.

Note that the idea is simply to encode all the memoryless randomised schedulers into the model (specifically into the actions). Formally one can show easily that

▶ Proposition 35.

$$\overline{H}(\mathcal{M}) = \overline{H}(\mathcal{M}') \text{ and } \overline{\nabla H}(\mathcal{M}) = \overline{\nabla H}(\mathcal{M}')$$

We then equip the semi-infinite MDP $\mathcal{M}'$ with a *reward structure*, i.e., to define $r : S \times Act' \to \mathbb{R}$ as $r(s,(x_t)_{t\in S}) = -\sum_{t\in S} x_t \log x_t$. Note that r is bounded. A simple but crucial observation is:

- by Proposition 14 and Theorem 8, the maximum entropy of $\mathcal{M}'$ is equal to the *maximal expected total reward* to reach G wrt. r (note that $r(s) = 0$ for $s \in G$); and similarly
- by Proposition 16 and Theorem 10, the maximum entropy rate of $\mathcal{M}'$ is equal to the *maximal limiting average reward* wrt. r;

This elegant link enables us to deduce that *simple schedulers* are sufficient for $\mathcal{M}'$ (which is a well-known fact, see, e.g., [26], Chapter 9 and Chapter 8-9 respectively), and furthermore, translating back to $\mathcal{M}$, memoryless *randomised* schedulers are sufficient to achieve the maximum entropy/entropy rate. We include a proof sketch for completeness.

▶ Proposition 36. Memoryless randomised schedulers suffice for MDPs to achieve maximum entropy/entropy rate.

Proof sketch. We shall follow standard argument in [26, 3] by considering the *state-action* frequency associated with the schedulers. For any state-action pair (s,a) such that $a \in \mathfrak{A}(s)$,

- for entropy, the frequency is a RV $X_{s,a}$ denoting the number of times the MC has taken (s,a) before reaching G under the scheduler σ;
- for entropy rate, the frequency is a RV $X_{s,a}$ denoting the limiting average of the number of times the MC has taken (s,a) in the long-run under the scheduler σ.

By Proposition 14 and Theorem 8 for entropy, and Proposition 16 and Theorem 10 for entropy rate, it is easy to observe that the entropy of $\mathcal{M}_\sigma$ depends only on the *expectation* of the corresponding state-action frequency and local entropy. The conclusion follows. ◀

▶ Remark. The maximum entropy problem for MDPs bears an interesting resemblance to the problem for IMCs. Indeed there is a link between IMCs and semi-infinite MDPs as well. Namely, we can associate an IMC $\mathcal{I} = (S, \alpha, \mathbf{P}^l, \mathbf{P}^u)$ with $\mathcal{M} = (S, \alpha, Act, \mathfrak{A}, \tau)$, where for each state $s \in S$,

$$\mathfrak{A}(s) = \{\mu \in \Delta(S) \mid \mathbf{P}^l(s,s') \leq \mu(s') \leq \mathbf{P}^u(s,s') \text{ for any } s' \in S\}$$

$Act = \bigcup_{s\in S} \mathfrak{A}(s)$, and $\tau(t|s,\mu) = \mu(t)$. We remark this is the *interval MDP* semantics of IMCs [14].

To start the link, we note that, for an IMC $\mathcal{I}$, the set of SCCs of $\mathcal{I}$ coincides to the set of MECs of the IMDP of $\mathcal{I}$, i.e. $\mathcal{M}$ defined above; this can be easily seen by inspecting Definition 32 and Definition 2.

Furthermore, it is rather straightforward to observe that the two semantics of IMCs, i.e., the UMC semantics and the IMDP semantics coincide for the maximum entropy/entropy

rate problem. (The crucial observation is that for IMDPs, simple schedulers are sufficient.) Hence essentially we establish the following link:

IMC with UMC semantics $\Leftrightarrow$ IMC with IMDP semantics $\Leftrightarrow$ semi-infinite MDP

In Section 5, we tackle the UMC semantics and indeed, most of the results/method will be carried over to MDPs readily.

Entropy. Again we assume that $\overline{H}(\mathcal{M}) < \infty$. Hence we can identify a set of $G \subseteq S$ as in the IMC case. The link to the expected total reward objective to $\mathcal{M}'$ gives the following non-linear program, which can be seen as the counterpart of (3).

$$\begin{aligned} \text{minimise} \quad & \sum_{s\in S} \alpha(s) y_s + \mathbb{H}(\alpha) \\ \text{subject to} \quad & y_s \geq \max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y}) \quad s \notin G \\ & y_s = 0 \qquad\qquad\qquad\quad s \in G \end{aligned} \tag{16}$$

where $\Omega(s)$ is defined as $\vec{x}_s = (x_{s,a})_{a\in\mathfrak{A}(s)}$

$$\vec{x}_s \in \Omega(s) \;\text{ iff }\; \sum_{a\in\mathfrak{A}(s)} x_{s,a} = 1 \text{ and } x_{s,a} \geq 0$$

and $\Gamma(\vec{x}_s, \vec{y}) = \sum_{t\in S} p(s,t) y_t - p(s,t) \log p(s,t)$ where $p(s,t) = \sum_{a\in\mathfrak{A}(s)} x_{s,a} \cdot \tau(t|s,a)$.

Entropy rate. As in the IMC, we start by identifying all the MECs of $\mathcal{M}$. The link to the limiting average reward objective to $\mathcal{M}'$ gives the following non-linear programs, which can be seen as the counterparts of (6) and (11). For each $B \in \mathcal{E}(\mathcal{M})$, we consider the following non-linear program. 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 $\vec{x}_s = (x_{s,t})_{t\in S}$.

$$\begin{aligned} \text{minimise} \quad & r \\ \text{subject to} \quad & r + y_s \geq \max_{\vec{x}_s \in \Omega(s)} \Gamma(\vec{x}_s, \vec{y}) \quad s \in B \end{aligned} \tag{17}$$

For each B, we obtain that r_B is the optimal value of (5). For simplicity, we define, for a given vector $\vec{z} = (z_s)_{s\in S}$, $\Lambda(\vec{x}_s, \vec{z}) = \sum_{t\in S} \sum_{a\in\mathfrak{A}(s)} x_{s,a} \tau(t|s,a) z_t$. We then consider the following non-linear program

$$\begin{aligned} \text{minimise} \quad & \sum_{s\in S} \alpha(s) z_s \\ \text{subject to} \quad & z_s \geq \max_{\vec{x}_s \in \Omega(s)} \Lambda(\vec{x}_s, \vec{z}) \qquad\qquad \forall s \in S \\ & z_s \geq r_B \qquad\qquad\qquad\quad \forall s \in S \text{ and } s \in B \end{aligned} \tag{18}$$

Threshold problem. The threshold problem can also be solved by slight adaptation alongside of Section 5.3. In particular, by the first-order theory, it sufffices to consider

$$\exists \vec{x}, \vec{y}. \bigwedge \begin{cases} \sum_{s\in S\setminus G} \alpha(s) \vec{y}_s \geq \theta \\ y_s = \sum_{t\in S} \sum_{a\in\mathfrak{A}(s)} x_{s,a} \tau(t|s,a) y_t \\ \qquad\quad - \sum_{t\in S} \sum_{a\in\mathfrak{A}(s)} x_{s,a} \tau(t|s,a) \log(\sum_{a\in\mathfrak{A}(s)} x_{s,a} \tau(t|s,a)) \quad \forall s \in S \setminus G \\ x_{s,a} \geq 0 \quad \forall s \in S \\ \sum_{a\in\mathfrak{A}(s)} x_{s,a} = 1 \quad \forall s \in S \end{cases}$$

where $\vec{x}$ is the concatenation of $\vec{x}_s = (x_{s,a})_{a\in\mathfrak{A}(s)}$ for $s \in S \setminus G$ and $\vec{y} = (y_s)_{s\in S}$.

And

$$\forall\vec{z}\exists\vec{x},\vec{y},\vec{r}.\bigwedge\begin{cases}\sum_{s\in S}\alpha(s)\vec{z}_s \geq \theta\\ z_s \geq \sum_{t\in S}\sum_{a\in\mathfrak{A}(s)} x_{s,a}\tau(t|s,a)z_t \quad \forall s\in S\\ z_s \geq r_B \quad \forall B\in\mathcal{E}(\mathcal{I})\\ r_B + z_s = \sum_{t\in S}\sum_{a\in\mathfrak{A}(s)} x_{s,a}\tau(t|s,a)z_t\\ \quad -\sum_{t\in S}\sum_{a\in\mathfrak{A}(s)} x_{s,a}\tau(t|s,a)\log(\sum_{a\in\mathfrak{A}(s)} x_{s,a}\tau(t|s,a)) \quad \forall B\in\mathcal{E}(\mathcal{M})\wedge s\in B\\ x_{s,a}\geq 0 \quad \forall s\in S\\ \sum_{a\in\mathfrak{A}(s)} x_{s,a} = 1 \quad \forall s\in S\end{cases}$$

where $\vec{x}$ is the concatenation of $\vec{x}_s = (x_{s,a})_{t\in S}$ for $s \in S$, $\vec{y} = (y_s)_{s\in S}$, $\vec{z} = (z_s)_{s\in S}$ and $\vec{r} = (r_B)_{B\in\mathcal{E}(\mathcal{I})}$.

With the same methods to solve (16), (18) and (17), and the same argument as in Section 5.3, we conclude:

▶ **Theorem 37.** *Given the MDP $\mathcal{M}$ and $\epsilon > 0$,*

1. *$\overline{H}(\mathcal{M})$ and $\overline{\nabla H}(\mathcal{M})$ can be approximated upper to ϵ in polynomial time in $\sharp\mathcal{M}$ and $\log(\frac{1}{\epsilon})$.*
2. *if the first-order theory of $(\mathbb{R},+,-,\cdot,e^x,0,1,\leq)$ is decidable, or if $\overline{H}(\mathcal{M}) \neq \theta$ (resp. $\overline{\nabla H}(\mathcal{M}) \neq \theta$), then $\overline{H}(\mathcal{M}) \bowtie \theta$ and $\overline{\nabla H}(\mathcal{M}) \bowtie \theta$ are decidable for $\bowtie \in \{\leq,<,=,>,\geq\}$.*

References

1 C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
2 S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
3 L. de Alfaro. Formal verification of probabilistic systems. PhD Thesis. Stanford University, 1997.