A Generalization of the Blahut–Arimoto Algorithm to Finite-State Channels

Pascal O. Vontobel, *Member, IEEE*, Aleksandar Kavčić, *Senior Member, IEEE*, Dieter M. Arnold, *Member, IEEE*, and Hans-Andrea Loeliger, *Fellow, IEEE*

***Abstract*—The classical Blahut–Arimoto algorithm (BAA) is a well-known algorithm that optimizes a discrete memoryless source (DMS) at the input of a discrete memoryless channel (DMC) in order to maximize the mutual information between channel input and output. This paper considers the problem of optimizing finite-state machine sources (FSMSs) at the input of finite-state machine channels (FSMCs) in order to maximize the mutual information rate between channel input and output. Our main result is an algorithm that efficiently solves this problem numerically; thus, we call the proposed procedure the generalized BAA. It includes as special cases not only the classical BAA but also an algorithm that solves the problem of finding the capacity-achieving input distribution for finite-state channels with no noise. While we present theorems that characterize the local behavior of the generalized BAA, there are still open questions concerning its global behavior; these open questions are addressed by some conjectures at the end of the paper. Apart from these algorithmic issues, our results lead to insights regarding the local conditions that the information-rate-maximizing FSMSs fulfill; these observations naturally generalize the well-known Kuhn–Tucker conditions that are fulfilled by capacity-achieving DMSs at the input of DMCs.**

***Index Terms*—Blahut–Arimoto algorithm (BAA), capacity, constrained capacity, finite-state machine channels (FSMCs), finite-state machine sources (FSMSs), information rate, optimization, run-length constraints.**

Manuscript received November 29, 2004; revised March 2, 2007. The work of P. O. Vontobel was supported in part by ETH under Grant TH-16./99-3 and by the National Science Foundation under Grants CCR 99-84515 and CCR 01-05719. The work of A. Kavčić was supported in part by the National Science Foundation under Grant CCR-0118701. The material in this paper was presented in part at IEEE Globecom, San Antonio, TX, Nov. 2001, and at the IEEE International Symposium on Information Theory, Pacifico Yokohama, Japan, June/July 2003.

P. O. Vontobel was with the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland, and with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL USA. He is now with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: pascal.vontobel@ieee.org).

A. Kavčić was with the Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA. He is now with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 USA (e-mail: kavcic@spectra.eng.hawaii.edu).

D. M. Arnold was with the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland, and with the IBM Zurich Research Laboratory, Rueschlikon, Switzerland. He is now with Siemens Switzerland AG, 8047 Zurich, Switzerland (e-mail: Dieter.M.Arnold @siemens.com).

H.-A. Loeliger is with the Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland (e-mail: loeliger @isi.ee.ethz.ch).

Communicated by Y. Steinberg, Associate Editor for Shannon Theory.

Digital Object Identifier 10.1109/TIT.2008.920243

I. Introduction

In this paper we consider the problem of computing the capacity of a finite-state machine channel (FSMC). An FSMC is a channel with memory whose channel characteristics are determined by one of finitely many states that the channel can be found in. The most abundant example of an FSMC is the partial response channel [1] that is found in magnetic and optical recording [2] as well as in communications over band-limited channels with intersymbol interference (ISI) when the input alphabet is constrained to be finite [3]. Other examples are the Gilbert–Elliott channel [4] and similar channels [5], where the state transitions are governed by a Markov chain. Many other channels that exhibit memory can be modeled (with a fair degree of accuracy) as FSMCs [6], [7]. Also, the computation of the capacity of constrained sequences (such as run-length-limited sequences) transmitted over channels with and without memory [8] can be formulated as a problem of computing the capacity of an FSMC.

The computation of the capacity of an FSMC has long been an open problem in information theory. In contrast, the computation of the capacity of a memoryless channel has long been solved. Shannon [9] computed the closed-form capacity of a memoryless additive white Gaussian noise channel under an average power constraint, and provided several closed-form solutions for simple discrete memoryless channels (DMCs), such as the binary symmetric channel. A general numeric procedure for computing the capacity of a general DMC was derived by Arimoto and Blahut [10], [11], hereafter called the classical Blahut-Arimoto algorithm (classical BAA). This method also applies to continuous-output memoryless channels (see [11, Sec. V]). Further, the classical BAA can be cast as a stochastic algorithm [12], [13].

For channels with memory, there exist several capacity computation methods. For Gaussian channels with ISI and an average power constraint, the capacity is computed by the water-filling theorem [14]–[16]. The capacity of Gilbert–Elliott-type finite-state channels is also known [4], [5]. However, the capacity of FSMCs that exhibit ISI (a prime example being the partial response channel) has remained a challenge [17].

The definition of the channel capacity C of a partial response channel (or more precisely, an FSMC) can be found in [15, p. 109]. Often authors refer to a different capacity, the independent and uniformly distributed (i.u.d.) capacity $C_{\text{i.u.d.}}$, which is defined as the information rate when the channel inputs are i.u.d. random variables. If the channel inputs are antipodal (i.e., ± 1), then $C_{\text{i.u.d.}}$ is also referred to as symmetric information rate [18].

Since the problem of computing the capacity C and the i.u.d. capacity $C_{\text{i.u.d.}}$ of a finite-state channel has not been solved, researchers typically reverted to computing bounds on these two capacities. A notable contribution toward the computation of the i.u.d. capacity of a partial response channel is Hirt's Ph.D. dissertation [18], where a Monte Carlo method is used to evaluate bounds on $C_{\text{i.u.d.}}$.[1] Subsequent efforts were concentrated at deriving analytic expressions for upper and lower bounds on C and $C_{\text{i.u.d.}}$. Shamai *et al.* [20], [21] derived upper and lower bounds. Further, a result by Shamai and Verdú [22] can be used as a lower bound [23], which is particularly tight at low signal-to-noise ratios (SNRs). Moreover, an interesting contribution is the Shamai–Laroia [23] *conjectured* lower bound on $C_{\text{i.u.d.}}$, which has still not been proved (nor disproved). As the capacities of partial response channels are closely connected to the information rates (and capacities) of finite-state machines and Markov chains, there are many closely related papers. These include entropy rates of constrained sequences and Markov chains [9], capacities with cost constraints [24], lower bounds on the noisy capacities of run-length-limited codes over memoryless channels [8], and results on transmission rates of Markov chains [25].

An efficient Monte Carlo method for computing the information rate of an FSMC whose input is a Markov process (including the i.u.d. capacity $C_{\text{i.u.d}}$) was proposed independently by Arnold and Loeliger [26], Sharma and Singh [27], Pfister *et al.* [28]. A summary of these methods as well as extensions to certain infinite-memory channels can be found in [19]. In the present paper, we are concerned with the computation (more precisely, the computation of tight lower bounds) of the actual capacity of finite-state channels. That is, we pose an optimization problem and propose an algorithmic solution, hereafter called the generalized Blahut–Arimoto algorithm (generalized BAA). Essentially, this algorithm is the one proposed in [12], with the difference that we now allow also noncontrollable channels (for a definition, see Remark 21 and Definition 22) and that we now prove specific statements regarding the (local) convergence of the proposed algorithm. (Note that in [29], [30] the results were restricted to the case of controllable channels.) Thus, the method presented here optimizes a Markov process of a certain (fixed) memory length to achieve the highest possible information rate over a finite-state channel under the Markov memory-length constraint. As the Markov memory is gradually increased, we start approaching the capacity of the channel arbitrarily closely.

Since the optimization method in this paper concerns only Markov (i.e., finite) memory sources, the optimized information rates are essentially lower bounds on the channel capacity. By virtue of comparison to upper bounds, we claim that our lower bounds are numerically tight. However, the constructions of the upper bounds are beyond the scope of this paper, and can be found in [31], [32]. (The closeness of the lower and upper bounds is not unexpected given that Chen and Siegel [33] have shown that as the Markov source memory goes to infinity, a Markov process can achieve asymptotically the unconstrained capacity of a finite-state ISI channel.)

[1] In [19], we comment on the relationship between Hirt's bounds and $C_{\text{i.u.d.}}$.

A. Organization

The paper is organized into the following sections.

- Section II discusses the classical BAA that finds a capacity-achieving input distribution of a discrete memoryless channel. It is presented in a way that will make the transition to the generalized BAA in Section IV transparent.
- Section III introduces the FSMC model and the necessary notation.
- The aim of Section IV is to introduce the generalized BAA as an extrapolation from the classical BAA [10], [11]. Thereby, we do not burden this section with proofs with the goal of making the section accessible to a wider audience.
- Section V carries the theoretical weight of this paper. This section contains a series of lemmas and theorems that build toward a (local) convergence proof of the proposed algorithm. The main result of the section is a lemma that allows us to claim that if numerical convergence of the algorithm is observed, then the resulting rate is at least a *local* maximum of the information rate for a Markov source.
- In Section VI, for several chosen FSMCs, we give results that numerically support the claim that the local maxima are very likely also global maxima. We show that the computed lower bounds are extremely tight by comparing them to upper bounds presented in [31], [32].
- In Section VII, we address what we believe to be important open problems that concern the convergence proof. In particular, we state a concavity conjecture, which, if proved, would guarantee that the presented algorithm cannot get stuck in (nonglobal) local maxima of the mutual-information-rate function because there are no such maxima. The proof of another concavity conjecture would yield the result that the proposed generalized BAA gives Markov sources that increase the mutual information rate after each step.
- Section VIII concludes the paper.

B. Notation

The following general notation will be used. Other notation will be introduced along the way.

- Alphabets will be denoted by calligraphic characters.
- Vectors will be denoted by boldface Greek characters. If $\boldsymbol{\gamma}$ is a vector, then the ith element of $\boldsymbol{\gamma}$ is denoted by γ_i. Matrices will be denoted by sans-serif boldface Latin letters. If $\mathbf{A}$ is a matrix, then A_{ij} denotes the element in the ith row and jth column.
- Random variables will be denoted by upper-case characters (e.g., X) while their realizations will be denoted by lower-case characters (e.g., x). Random vectors will be denoted by upper-case boldface characters (e.g., $\boldsymbol{X}$) while their realizations will be denoted by lower-case boldface characters (e.g., $\boldsymbol{x}$).
- The ℓth member of a sequence $\{x\}$ is denoted by x_ℓ. If $i \leq j$ then $\boldsymbol{x}_i^j$ denotes the vector $[x_i, x_{i+1}, \ldots, x_{j-1}, x_j]$; otherwise, $\boldsymbol{x}_i^j$ denotes the empty vector.
- The probability of an event $\mathcal{E}$ is denoted by $\Pr(\mathcal{E})$. The conditional probability of an event $\mathcal{E}_1$ given an event $\mathcal{E}_2$ is denoted by $\Pr(\mathcal{E}_1|\mathcal{E}_2)$.

- All logarithms are natural logarithms (base e); therefore, all entropies and mutual informations will be measured in nats. The only exceptions are figures where the information rates will be plotted in bits per channel use.
- For the purpose of developing the theory, all input, output, and state alphabets will be assumed to be finite; therefore, when talking about probabilities, we will only talk about probability mass functions (pmfs). Basically, under suitable conditions all the results can be extended to the case where the output alphabet is $\mathbb{R}$; in this case, the corresponding pmfs and sums must be changed to probability density functions (pdfs) and integrals, respectively. (Reference [19] considers some of these generalizations when computing information rates.) Note that in Section VI, where we give concrete computation examples, we do consider a case where the output alphabet is $\mathbb{R}$.
- In order to not clutter the summation signs too much we will use the following conventions (see also the notations in Definitions 15 and 27). Summations like $\sum_x$, $\sum_y$, $\sum_s$, and $\sum_b$ will implicitly mean $\sum_{x\in\mathcal{X}}$, $\sum_{y\in\mathcal{Y}}$, $\sum_{s\in\mathcal{S}}$, and $\sum_{b\in\mathcal{B}}$, respectively. Summations like $\sum_{\boldsymbol{x}}$ and $\sum_{\boldsymbol{y}}$ will implicitly mean $\sum_{\boldsymbol{x}\in\mathcal{X}_{-N+1}^{N}}$ and $\sum_{\boldsymbol{y}\in\mathcal{Y}_{-N+1}^{N}}$. Summations like $\sum_{\boldsymbol{s}}$ and $\sum_{\boldsymbol{b}}$ will be over all valid state and branch sequences of a trellis, respectively. (Trellises, and related notions like states, branches, state sequences, branch sequences, etc., will be formally introduced in Section III.)
- In summations like $\sum_{\boldsymbol{b}} Q(\boldsymbol{b}) f(Q(\boldsymbol{b}))$ (where $f : \mathbb{R} \to \mathbb{R}$ is some function, typically the logarithm function) we sum only over legal sequences $\boldsymbol{b}$ where $Q(\boldsymbol{b})$ is nonzero (i.e., only over $\boldsymbol{b}$'s that are in the support of $Q(\cdot)$). Similar conventions will be used for other random variables and vectors (see also Definition 27). Note that in order to keep the notation brief, in the following we will write Q instead of $Q(\cdot)$, W instead of $W(\cdot|\cdot)$, etc.
- Some quantities will be implicitly defined through other quantities (see also Remark 24). *E.g.*, if $\boldsymbol{b}$ is a branch sequence then $s_\ell \triangleq s_\ell(\boldsymbol{b})$ will denote the state at time ℓ that is visited by the branch sequence $\boldsymbol{b}$. In this case, we say that $\boldsymbol{b}$ and s_ℓ are compatible.
  Similarly, the state sequence $\boldsymbol{s} \triangleq \boldsymbol{s}(\boldsymbol{b})$ will denote the sequence of states that is visited by $\boldsymbol{b}$ and we will say that $\boldsymbol{s}$ and $\boldsymbol{b}$ are compatible. Obviously, if $\boldsymbol{b}$ is a legal branch sequence then $\boldsymbol{s}$ is a legal state sequence. In this spirit, summations like $\sum_{\boldsymbol{b}:s_\ell=i} f(\boldsymbol{b})$ will mean that we sum $f(\boldsymbol{b})$ over all legal branch sequences $\boldsymbol{b}$ where $s_\ell = s_\ell(\boldsymbol{b}) = i$. Moreover, summations like $\sum_{\boldsymbol{b}} f(\boldsymbol{b}) \sum_{\boldsymbol{b}'} g(\boldsymbol{b}'')$ will mean that the second summation is over all valid $\boldsymbol{b}''$ that are consistent with $\boldsymbol{b}$ from the first summation.
- The symbol α will have a special meaning in Section V, see Definition 51. There we will consider input distributions that are parameterized by a scalar α. The meaning of notations like $Q_{ij}^{\alpha}(\alpha)$ will also be introduced in Definition 51.

II. The Classical BAA for DMCs

This section is about finding a capacity-achieving input distribution to a DMC [16]. To this end, we will first review the definition of a DMC in Section II-A and define its capacity in Section II-B. The classical BAA is a well-known algorithm to obtain a capacity-achieving input distribution: Section II-C shows the main idea about this algorithm, whereas Section II-D gives a detailed description of it. The goal of this section is to present the classical BAA in a way that will make the step to the generalized BAA for finite-state channels transparent.

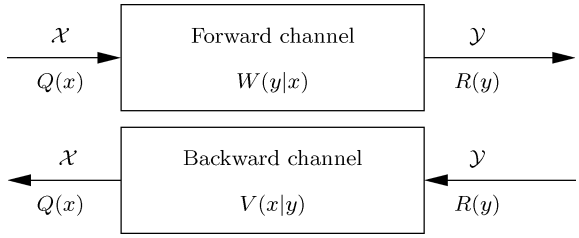

Fig. 1. DMC with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$. The "forward" channel law is given by $W(y|x)$. If the input has pmf $Q(x)$, the output has pmf $R(y) = (QW)(y)$. The "backward" channel has the channel law $V(x|y)$.

A. DMCs

We consider a DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and channel law (conditional pmf)

$$W(y|x) \triangleq P_{Y|X}(y|x),$$

see Fig. 1. By definition, the alphabets are finite, i.e., $|\mathcal{X}| < \infty$ and $|\mathcal{Y}| < \infty$. We let the channel input be a random variable X over $\mathcal{X}$, and we denote its pmf by $Q(x) \triangleq P_X(x)$. The channel output is correspondingly a random variable Y over $\mathcal{Y}$ with pmf

$$R(y) \triangleq (QW)(y) \triangleq P_Y(y) = \sum_x Q(x)W(y|x).$$

The *a posteriori* probability of $X = x$ upon observing $Y = y$ shall be denoted by

$$V(x|y) \triangleq P_{X|Y}(x|y) = \frac{Q(x)W(y|x)}{R(y)} = \frac{Q(x)W(y|x)}{(QW)(y)} = \frac{Q(x)W(y|x)}{\sum_{\overline{x}\in\mathcal{X}} Q(\overline{x})W(y|\overline{x})}.$$

The joint density of X and Y is therefore

$$P_{X,Y}(x,y) = Q(x)W(y|x) = R(y)V(x|y).$$

This yields the important relationship

$$\frac{V(x|y)}{Q(x)} = \frac{W(y|x)}{R(y)}$$

between $Q(x)$, $W(y|x)$, $R(y)$, and $V(x|y)$. Because

$$Q(x) = \sum_y R(y)V(x|y)$$

we can consider $V(x|y)$ to be a *backward* channel law (see Fig. 1).

In the following, we will assume that the channel law $W(y|x)$ is *fixed*, whereas the channel input distribution $Q(x)$ will be *varied*. However, note that varying $Q(x)$ will, of course, imply that also $R(y)$ and $V(x|y)$ *vary*! In other words, with a pmf $R(y)$ and a conditional pmf $V(x|y)$ there is implicitly a pmf $Q(x)$ behind them. Usually, we will try to make this clear by using some decorations on R and V. So, if the input pmf of X is $\tilde{Q}(x)$, then we denote the pmf of Y by $\tilde{R}(\cdot)$ and the *a posteriori*

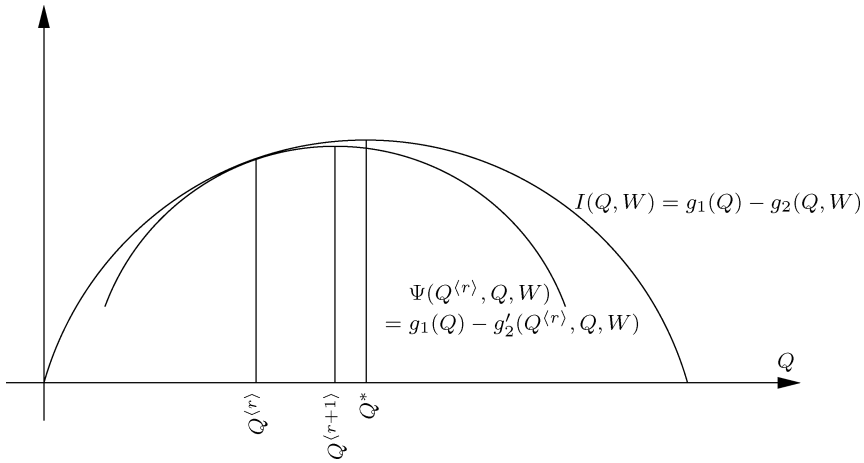

Fig. 2. Generic mutual information $I(Q,W)$ and surrogate function $\Psi(Q^{\langle r\rangle}, Q, W)$. Q^* is a capacity-achieving input distribution.

probability of $X = x$ upon observing $Y = y$ is called $\tilde{V}(x|y)$. We have the relations

$$\begin{aligned}\tilde{R}(y) &\triangleq (\tilde{Q}W)(y) \triangleq \sum_x \tilde{Q}(x)W(y|x)\\ \tilde{V}(x|y) &\triangleq \frac{\tilde{Q}(x)W(y|x)}{\tilde{R}(y)} = \frac{\tilde{Q}(x)W(y|x)}{(\tilde{Q}W)(y)}\\ &= \frac{\tilde{Q}(x)W(y|x)}{\sum_{\overline{x}\in\mathcal{X}} \tilde{Q}(\overline{x})W(y|\overline{x})}\\ \tilde{Q}(x)W(y|x) &= \tilde{R}(y)\tilde{V}(x|y). \qquad (1)\end{aligned}$$

Note that always $\tilde{W}(y|x) = W(y|x)$ since the channel law does not change (by definition), but $\tilde{V}(x|y) \neq V(x|y)$ in general.

B. Channel Capacity of a DMC

The following definitions are standard (see, e.g., [16]).

Definition 1 (Set $\mathcal{Q}$): We let $\mathcal{Q}$ be the set of all pmfs over $\mathcal{X}$, i.e.,

$$\mathcal{Q} \triangleq \left\{ Q : \mathcal{X} \to \mathbb{R} \;\middle|\; \begin{array}{c} Q(x) \geq 0 \text{ for all } x \in \mathcal{X} \\ \sum_x Q(x) = 1 \end{array} \right\}.$$

Definition 2 (Mutual Information): Let X and Y have the joint pmf $P_{X,Y}(x,y) = Q(x)W(y|x)$. The mutual information between X and Y is defined as

$$\begin{aligned}I(Q,W) &\triangleq I(X;Y)\\ &\triangleq H(Y) - H(Y|X)\\ &= \sum_x \sum_y Q(x)W(y|x) \log\left(\frac{W(y|x)}{(QW)(y)}\right)\\ &= H(X) - H(X|Y)\\ &= \sum_x \sum_y Q(x)W(y|x) \log\left(\frac{V(x|y)}{Q(x)}\right).\end{aligned}$$

Definition 3 (Channel Capacity): Let the DMC with input X and output Y have the channel law $W(y|x)$. The channel capacity is then defined as

$$C(W) \triangleq \max_{Q\in\mathcal{Q}} I(Q,W).$$

A pmf $Q \in \mathcal{Q}$ that maximizes $I(Q,W)$ is called a *capacity-achieving input distribution*. (Note that there are DMCs for which there is no unique capacity-achieving input distribution. However, note that one can show that for any DMC, all the output distributions induced by capacity-achieving input distributions are equal, see e.g., [34], [35].)

C. The Main Idea Behind the Classical BAA

The classical BAA [10], [11] (see also the tutorial [36]) solves the problem of numerically computing both the capacity and a capacity-achieving input distribution for a given DMC. In the following, we assume to have a fixed DMC with channel law $W(y|x)$. Fig. 2 schematically depicts a possible information rate $I(Q,W)$ as a function of Q. As the alphabet size $\mathcal{X}$ is usually at least two, the optimization problem is a multidimensional one. For illustration purposes, though, a one-dimensional representation of Q will do. The problem of finding a capacity-achieving input distribution is therefore to find where $I(\cdot,W)$ has a maximum. The problem is simplified by the fact that $I(Q,W)$ is concave in Q (see e.g., [16]).

There are of course different ways to find such a maximum. One of them would be to introduce Lagrangian multipliers for the constraints, formulate the Kuhn–Tucker conditions, and solve the resulting equation system; but this equation system is usually highly nonlinear. Other approaches leading to our goal would be gradient-based methods or interior-point algorithms. But a particularly elegant and efficient way to solve the problem at hand is the classical BAA. As it is a "nice" algorithm, there are many ways to describe it; we will choose a description that will ease the transition to the generalized BAA in Sections IV and V.

The main idea of the classical BAA is the following. It is an iterative algorithm, so assume that at iteration r we have found some input pmf $Q^{\langle r\rangle}$ with corresponding information rate $I(Q^{\langle r\rangle}, W)$ (see Fig. 2). At iteration $r+1$ we would like to find a "better" $Q^{\langle r+1\rangle}$, i.e., an input pmf for which $I(Q^{\langle r+1\rangle}, W) \geq I(Q^{\langle r\rangle}, W)$ (see Fig. 2). To this end, we introduce a surrogate function $\Psi(Q^{\langle r\rangle}, Q, W)$ which locally (i.e., around $Q = Q^{\langle r\rangle}$) approximates $I(Q,W)$ (see Fig. 2). We require the following:

- that the surrogate function assumes the same value at $Q = Q^{\langle r\rangle}$ as $I(Q,W)$ does, i.e., $\Psi(Q^{\langle r\rangle}, Q^{\langle r\rangle}, W) = I(Q^{\langle r\rangle}, W)$;
- that $\Psi(Q^{\langle r\rangle}, Q, W)$ is never above $I(Q,W)$, i.e., $\Psi(Q^{\langle r\rangle}, Q, W) \leq I(Q,W)$ for all Q;

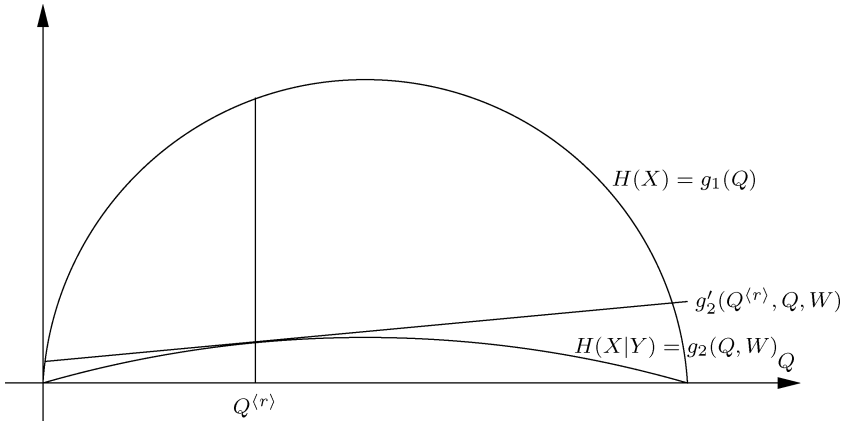

Fig. 3. Generic entropy $H(X) = g_1(Q)$ and conditional entropy $H(X|Y) = g_2(Q, W)$. $g_2'(Q^{\langle r \rangle}, Q, W)$ is a linear approximation of $H(X|Y)$ at $Q = Q^{\langle r \rangle}$.

- that maximizing $\Psi(Q^{\langle r \rangle}, Q, W)$ over Q is easy, i.e., it can be done in a computationally efficient way.

Assume that such a surrogate function can be found. Letting $Q^{\langle r+1 \rangle}$ be the pmf where $\Psi(Q^{\langle r \rangle}, Q, W)$ achieves its maximum over Q, i.e.,

$$Q^{\langle r+1 \rangle} \triangleq \arg\max_{Q \in \mathcal{Q}} \Psi\left(Q^{\langle r \rangle}, Q, W\right)$$

then $Q^{\langle r+1 \rangle}$ represents a new input pmf which is not only efficiently computable based on $Q^{\langle r \rangle}$ but which also fulfills $I(Q^{\langle r+1 \rangle}, W) \geq I(Q^{\langle r \rangle}, W)$ (see Fig. 2). (More about surrogate functions and their use can, e.g., be found in [37].)

There are different ways to motivate the surrogate function $\Psi(Q^{\langle r \rangle}, Q, W)$ that is used by the classical BAA. (We choose to show a construction that can be generalized later on when we will be talking about FSMCs.) We start by expressing $I(Q, W)$ as

$$\begin{aligned} I(Q, W) &= I(X; Y) = H(X) - H(X|Y) \\ &= g_1(Q) - g_2(Q, W) \end{aligned} \tag{2}$$

with

$$g_1(Q) \triangleq H(X) = -\sum_x Q(x) \log(Q(x))$$

$$\begin{aligned} g_2(Q, W) &\triangleq H(X|Y) \\ &= -\sum_x Q(x) \sum_y W(y|x) \log\left(\frac{Q(x)W(y|x)}{(QW)(y)}\right). \end{aligned}$$

Choosing

$$\Psi(Q^{\langle r \rangle}, Q, W) \triangleq g_1(Q) - g_2'(Q^{\langle r \rangle}, Q, W)$$

where g_2' is some function such that[2]

- $g_2'(Q^{\langle r \rangle}, Q, W)$ equals $g_2(Q, W)$ at $Q = Q^{\langle r \rangle}$, i.e., $g_2'(Q^{\langle r \rangle}, Q^{\langle r \rangle}, W) = g_2(Q^{\langle r \rangle}, W)$, and
- $g_2'(Q^{\langle r \rangle}, Q, W)$ is never below $g_2(Q, W)$, i.e., $g_2'(Q^{\langle r \rangle}, Q, W) \geq g_2(Q, W)$ for all Q,

leads to a function $\Psi(Q^{\langle r \rangle}, Q, W)$ that fulfills the desired requirements. By the concavity of $g_2(Q, W)$ in Q (which can be shown easily), such a function $g_2'(Q^{\langle r \rangle}, Q, W)$ can be chosen to be the linear approximation of $g_2(Q, W)$ at $Q^{\langle r \rangle}$, i.e., the function that goes through $g_2(Q, W)$ at $Q = Q^{\langle r \rangle}$ and that is tangential to $g_2(Q, W)$ (see Fig. 3). This is the approach taken by the classical BAA.

Doing the above iterations repeatedly not only leads to input pmfs where the mutual information gets potentially larger at each iteration, but for $r \to \infty$ the input pmf $Q^{\langle r \rangle}$ converges to a capacity-achieving input distribution (see Theorem 10).

D. Description of the Classical BAA

After having given the main idea behind the classical BAA for DMCs in Section II-C, we proceed to give the exact algorithm. Instead of introducing $g_2'(\tilde{Q}, Q, W)$ and showing that it fulfills the required properties as formulated in Section II-C, we will directly introduce $\Psi(\tilde{Q}, Q, W)$ and state its properties. To that end, it is useful to introduce the function $T(x)$.

Definition 4 (Function $T(x)$): We assume to have a DMC with a fixed channel law $W(y|x)$. If the input pmf is $Q(x)$ we define

$$\begin{aligned} T(x) &\triangleq \sum_y W(y|x) \log\big(V(x|y)\big) \\ &= \sum_y W(y|x) \log\left(\frac{Q(x)W(y|x)}{(QW)(y)}\right) \quad (\text{for all } x \in \mathcal{X}\,). \end{aligned}$$

If a different input pmf is used, we will decorate the symbol T. For example, if $\tilde{Q}(x)$ is the input pmf, we will have

$$\begin{aligned} \tilde{T}(x) &\triangleq \sum_y W(y|x) \log\left(\tilde{V}(x|y)\right) \\ &= \sum_y W(y|x) \log\left(\frac{\tilde{Q}(x)W(y|x)}{(\tilde{Q}W)(y)}\right) \quad (\text{for all } x \in \mathcal{X}\,). \end{aligned}$$

(Note that $T(x)$ and $\tilde{T}(x)$ are always nonpositive quantities.)

The quantity $T(x)$ can be seen as a measure for the "quality of the input symbol x" in the following sense. Assume x was sent and we observe the channel output. Then, the larger $T(x)$ is, the larger is the probability of observing a channel output value for which we can say with high likelihood that x was indeed sent.

[2]Note that the prime in the function label g_2' does not denote the derivative of g_2, it is merely used in order to introduce a function that is different from but closely related to g_2.

Let us note that with this definition of $T(x)$, the mutual information can be expressed as

$$I(Q,W) = \sum_x Q(x) \sum_y W(y|x) \log\left(\frac{V(x|y)}{Q(x)}\right) = \sum_x Q(x)\left[\log\left(\frac{1}{Q(x)}\right) + T(x)\right].$$

Definition 5 (Function Ψ): We assume to have a DMC with a fixed channel law $W(y|x)$. Let $\tilde{V}(x|y)$ for a given $\tilde{Q}(x)$ be defined as in (1). As hinted in Section II-C, the surrogate function $\Psi(\tilde{Q},Q,W)$ is defined as

$$\Psi(\tilde{Q},Q,W) = \sum_x Q(x) \sum_y W(y|x) \log\left(\frac{\tilde{V}(x|y)}{Q(x)}\right) = \sum_x Q(x)\left[\log\left(\frac{1}{Q(x)}\right) + \tilde{T}(x)\right]$$

with $\tilde{T}(x)$ as introduced in Definition 4.

Lemma 6 (Properties of Ψ): For all Q, $\tilde{Q}$, and W we have $\Psi\big(\tilde{Q},Q,W\big) \le \Psi\big(Q,Q,W\big)$ and $\Psi\big(Q,Q,W\big) = I(Q,W)$. Moreover, given a channel law W and some source pmf $\tilde{Q}$ there always exists a Q such that $I(\tilde{Q},W) \le \Psi\big(\tilde{Q},Q,W\big) \le I(Q,W)$.

Proof: Observe that

$$\Psi(\tilde{Q},Q,W) = I(Q,W) - \sum_y R(y)\left[\sum_x V(x|y)\log\left(\frac{V(x|y)}{\tilde{V}(x|y)}\right)\right].$$

The result is then a consequence of the well-known properties of relative entropy [16]; we omit the details. □

Remark 7 (Connection to the Outline in Section II-C): With the notation of Definition 5, the definitions in Section II-C are

$$\begin{aligned}
\tilde{Q} &= Q^{\langle r\rangle},\\
g_1(Q) &= -\sum_x Q(x)\log(Q(x)),\\
g_2(Q,W) &= -\sum_x Q(x)\sum_y W(y|x)\log(V(x|y)),\\
g_2'(\tilde{Q},Q,W) &= -\sum_x Q(x)\sum_y W(y|x)\log(\tilde{V}(x|y))\\
&= -\sum_x Q(x)\tilde{T}(x).
\end{aligned}$$

That $g_2'(\tilde{Q},Q,W)$ is tangential to $g_2(\tilde{Q},W)$ at $Q = \tilde{Q}$ can be shown in the following way. Assume to have a family of input pmfs that is parameterized by α. More specifically, we assume that for each $x \in \mathcal{X}$, $Q(x)$ is a smooth function of some parameter α. Additionally, we assume that $Q(x) = \tilde{Q}(x)$ for all $x \in \mathcal{X}$ when $\alpha = \tilde{\alpha}$, where $\tilde{\alpha}$ is some constant. Then one can show that

$$\begin{aligned}
g_2'(\tilde{Q},Q,W)\Big|_{\alpha=\tilde{\alpha}} &= g_2(Q,W)|_{\alpha=\tilde{\alpha}},\\
\frac{\mathrm{d}}{\mathrm{d}\alpha} g_2'(\tilde{Q},Q,W)\Big|_{\alpha=\tilde{\alpha}} &= \frac{\mathrm{d}}{\mathrm{d}\alpha} g_2(Q,W)\Big|_{\alpha=\tilde{\alpha}}\\
&= -\sum_x \left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(x)\right)\tilde{T}(x)\Big|_{\alpha=\tilde{\alpha}}.
\end{aligned}$$

Because $g_2'(\tilde{Q},Q,W)$ is linear in $Q(x)$, $g_2'(\tilde{Q},Q,W)$ is a linear approximation of $g_2(\tilde{Q},W)$ at $Q = \tilde{Q}$.[3]

Lemma 8 (Maximizing Ψ): Let

$$Q^* = \arg\max_{Q\in\mathcal{Q}} \Psi\big(\tilde{Q},Q,W\big)$$

then

$$Q^*(x) = \frac{\mathrm{e}^{\tilde{T}(x)}}{\sum_{\bar{x}} \mathrm{e}^{\tilde{T}(\bar{x})}}.$$

Proof: The proof can be obtained by the method of Lagrange multipliers. Alternatively, observe that

$$\Psi(\tilde{Q},Q,W) = -\sum_x Q(x)\log\left(\frac{Q(x)}{\mathrm{e}^{\tilde{T}(x)}/\sum_{\bar{x}}\mathrm{e}^{\tilde{T}(\bar{x})}}\right) + \log\left(\sum_{\bar{x}}\mathrm{e}^{\tilde{T}(\bar{x})}\right).$$

The result is then a consequence of $\mathrm{e}^{\tilde{T}(x)} \ge 0$ for all $x \in \mathcal{X}$ and the well-known properties of relative entropy [16]; we omit the details. □

Algorithm 9 (Classical BAA): We consider a DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and channel law $W(y|x)$. Let $Q^{\langle 0\rangle}$ be some initial (freely chosen) input distribution, the only requirement being $Q^{\langle 0\rangle}(x) > 0$ for all $x \in \mathcal{X}$. For iterations $r = 0,1,2,\ldots$ perform alternatively the following two steps.

- **First Step:** For each $x \in \mathcal{X}$ calculate

$$T^{\langle r\rangle}(x) = \sum_y W(y|x)\log\left(V^{(r)}(x|y)\right) = \sum_y W(y|x)\log\left(\frac{Q^{\langle r\rangle}(x)W(y|x)}{(Q^{\langle r\rangle}W)(y)}\right).$$

- **Second Step:** For each $x \in \mathcal{X}$, the new input probability $Q^{\langle r+1\rangle}(x)$ is calculated according to

$$Q^{\langle r+1\rangle}(x) = \frac{\mathrm{e}^{T^{\langle r\rangle}(x)}}{\sum_{\bar{x}}\mathrm{e}^{T^{\langle r\rangle}(\bar{x})}}.$$

Theorem 10 (Properties of the Classical BAA): For each $r = 0,1,2,\ldots$, the sequence of $Q^{\langle r\rangle}$ of input distributions produced by the classical BAA fulfills

$$I\left(Q^{\langle r+1\rangle},W\right) \ge I\left(Q^{\langle r\rangle},W\right).$$

Furthermore, $Q^{\langle r\rangle}$ converges to a capacity-achieving input distribution for $r \to \infty$.

Proof: The proof is a classic result [10], [11] and is omitted. See also the proofs in [36], [38]. □

Lemma 11 (Capacity Upper and Lower Bounds): Let $C = C(W)$ be the capacity for a given DMC with channel law $W(y|x)$. For any input pmf Q we have

$$\min_{\substack{x\\ Q(x)>0}} \big[T(x) - \log(Q(x))\big] \le I(Q,W) \le C \le \max_{\substack{x\\ Q(x)>0}} \big[T(x) - \log(Q(x))\big]$$

[3]We could even allow an additive constant in the definition of $g_2'(\bar{Q},Q,W)$; the function would still be a linear (or, more precisely, an affine) approximation.

Fig. 4. Normal factor graph representing the joint pmf (conditioned on s_{-N}) of a finite-state machine source model.

where $T(x)$ is as defined in Definition 4. For a capacity-achieving input pmf $Q(x)$, all inequalities turn into equalities.

Proof: Omitted. See also [15, Problem 4.17 on p. 524f.]. □

Remark 12 (Termination Condition for the Classical BAA): From Lemma 11 we see that we can take the quantity

$$\max_{\substack{x \\ Q(x)>0}} \big[T(x) - \log(Q(x))\big] - I(Q,W)$$

as a measure of how close we are to capacity. We can also take the quantity

$$\max_{\substack{x \\ Q(x)>0}} \big[T(x) - \log(Q(x))\big] - \min_{\substack{x \\ Q(x)>0}} \big[T(x) - \log(Q(x))\big]$$

for this purpose. Note that already before the introduction of the classical BAA, Gallager [15, Problem 4.17 on p. 524f.] proposed these search termination criteria.

III. Finite-State Machine Source Models and Finite-State Machine Channel Models: Definitions and Capacities

Whereas Section II-A discussed DMCs, this section takes a look at a much broader class of channels, namely indecomposable FSMCs as defined by Gallager [15]. These types of channels are characterized by the fact that the new state and the current output symbol are *stochastically* dependent on the previous channel state and the current channel input symbol.

It is well known that memoryless input distributions achieve the capacity of DMCs. But for channels with memory this is not the case anymore in general: in order to get higher information rates it is necessary to consider sources with memory. Therefore, this section will also study a class of sources with memory that can be described with indecomposable finite-state machine source (FSMS) models.

The purpose of Sections IV and V will then be to give an algorithm that finds the mutual-information-rate-maximizing parameters of an indecomposable FSMS at the input to an indecomposable FSMC.

This section is structured as follows: Section III-A defines FSMSs, Section III-B defines FSMCs, and Section III-C considers finite-state machines that describe an FSMS and an FSMC jointly. Whereas Section III-D looks at the unconstrained channel capacity of FSMCs, Section III-E considers the channel capacity for FSMCs that have an FSMS at the input.

A. FSMSs

Before turning to the definition of FSMSs, it is useful to define some index sets.

Definition 13 (Some Useful Index Sets): We assume N to be a positive integer; note that in all our results we will mainly be interested in the limit $N \to \infty$. We will use the index sets

$$\begin{aligned}\mathcal{I}_N &\triangleq [-N+1, N] &&= \{-N+1, \ldots, N\}\\ \mathcal{I}'_N &\triangleq [-N+1, N-1] &&= \{-N+1, \ldots, N-1\}.\end{aligned}$$

Observe that $|\mathcal{I}_N| = 2N$ and that $|\mathcal{I}'_N| = 2N-1$.

Definition 14 (FSMSs): A time-invariant (discrete-time) FSMS has a state sequence $\ldots, S_{-1}, S_0, S_1, \ldots$ and an output sequence $\ldots, X_{-1}, X_0, X_1, \ldots$ where $S_\ell \in \mathcal{S}$ and $X_\ell \in \mathcal{X}$ for all $\ell \in \mathbb{Z}$. We assume that the alphabets $\mathcal{X}$ and $\mathcal{S}$ are finite and that for any $N > 0$ the joint pmf decomposes as

$$\begin{aligned}P_{\boldsymbol{S}^N_{-N+1}, \boldsymbol{X}^N_{-N+1}|S_{-N}}(\boldsymbol{s}^N_{-N+1}, \boldsymbol{x}^N_{-N+1}|s_{-N})\\ = \prod_{\ell \in \mathcal{I}_N} P_{S_\ell, X_\ell|S_{\ell-1}}(s_\ell, x_\ell|s_{\ell-1}),\end{aligned}$$

where $P_{S_\ell, X_\ell|S_{\ell-1}}$ is independent of ℓ. This factorization is shown in the normal factor graph[4] in Fig. 4.

It is useful to introduce the random variable $B_\ell \triangleq (S_{\ell-1}, X_\ell, S_\ell)$; then, $P_{B_\ell}(b_\ell)$ represents the probability of choosing branch $b_\ell = \big(s_{\ell-1}(b_\ell), x_\ell(b_\ell), s_\ell(b_\ell)\big)$ at time index ℓ, i.e., the probability to be in state $s_{\ell-1}(b_\ell)$ at time index $\ell-1$, to choose symbol $x_\ell(b_\ell)$ at time index ℓ, and to be in state $s_\ell(b_\ell)$ at time index ℓ. Moreover, we will use the notation

$$\begin{aligned}Q(s_\ell, x_\ell|s_{\ell-1}) &\triangleq P_{S_\ell, X_\ell|S_{\ell-1}}(s_\ell, x_\ell|s_{\ell-1})\\ &= P_{B_\ell|S_{\ell-1}}\big((s_{\ell-1}, x_\ell, s_\ell)|s_{\ell-1}\big).\end{aligned}$$

We will only consider sources where the pair $(s_{\ell-1}, x_\ell)$ determines the pair $(s_{\ell-1}, s_\ell)$. With the exception of Section IV-C (which explains how to treat parallel branches), we will actually make an even more restrictive assumption, namely, that there is a one-to-one relationship between the pairs $(s_{\ell-1}, x_\ell)$ and $(s_{\ell-1}, s_\ell)$ (this excludes parallel branches in the trellis). From this follows that there is a one-to-one relationship between $(s_{-N}, \boldsymbol{x}^N_{-N+1})$ and $\boldsymbol{s}^N_{-N}$.

Definition 15 (FSMS Notation): The internal details of an FSMS (defined as in Definition 14 and depicted by a factor graph as in Fig. 4) can be visualized by a trellis as, e.g., shown in Fig. 5. (Note that showing a single trellis section is actually sufficient because of the assumed time invariance.) Focusing on this exemplary trellis, we introduce the notation that will be used throughout this paper. (We remind the reader of the simplifying assumption that we made in the second half of Definition 14.)

[4]For a definition of factor graphs in general, see [39]; for normal factor graphs in particular, see [40], [41]. One simply has to draw a circle on each edge of a normal factor graph in order to obtain a (standard) factor graph.

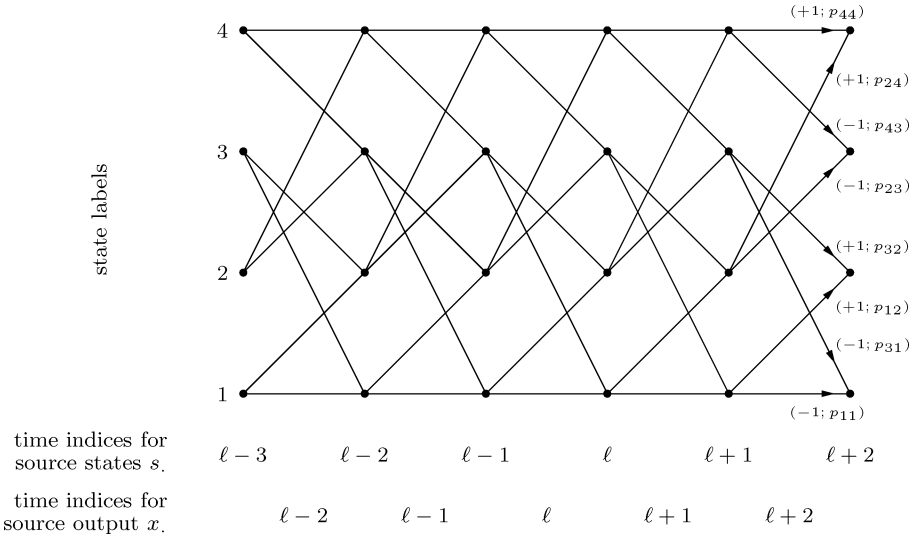

Fig. 5. Trellis representation of an FSMS. (The meaning of $(x_{ij}; p_{ij})$ is explained in Definition 15.)

- The state alphabet is $\mathcal{S}$ with state size $|\mathcal{S}|$ and the source output alphabet is $\mathcal{X}$ of size $|\mathcal{X}|$. In our example we have $|\mathcal{S}| = 4$ with $\mathcal{S} = \{1,2,3,4\}$ and an output alphabet of size $|\mathcal{X}| = 2$ with $\mathcal{X} = \{-1,+1\}$.
- Let the set $\mathcal{B}$ contain all triples (i, x_{ij}, j) that constitute legal branches, i.e., where the branch probabilities are allowed to be nonzero. Under the simplifying assumptions made in the second half of Definition 14, it is sufficient to consider the simplified set $\mathcal{B}$ that consists of all pairs (i,j) that constitute legal state transitions, i.e., where the transition probabilities are allowed to be nonzero. (In this case, x_{ij} is automatically defined through i and j.) From the context it will be clear if the original or the simplified version is used. In our example we have

$$\mathcal{B} = \{(1,-1,1),(1,+1,2),(2,-1,3),(2,+1,4),\\ (3,-1,1),\ (3,+1,2),(4,-1,3),(4,+1,4)\}$$

or the simplified

$$\mathcal{B}=\{(1,1),(1,2),(2,3),(2,4),(3,1),(3,2),(4,3),(4,4)\}.$$

- Let $\overrightarrow{\mathcal{B}_i} \triangleq \{j \,|\, (i,j) \in \mathcal{B}\}$ be the set of all legal follow-up states of state i, and let $\overleftarrow{\mathcal{B}_i} \triangleq \{k \,|\, (k,i) \in \mathcal{B}\}$ be the set of all legal preceding states of state i. (In our example we have $\overrightarrow{\mathcal{B}_1} = \{1,2\}$, $\overleftarrow{\mathcal{B}_1} = \{1,3\}$, etc.)
- To a legal state transition $(i,j) \in \mathcal{B}$ we associate a label $(x_{ij}; p_{ij})$ showing the source output symbol $x_{ij} \in \mathcal{X}$ and the transition probability p_{ij} of going from state i to state j, i.e., $p_{ij} = \Pr(S_\ell = j, X_\ell = x_{ij} | S_{\ell-1} = i) = \Pr(S_\ell = j | S_{\ell-1} = i)$ for all ℓ.
- Because all transition probabilities are assumed to be *time invariant* it makes sense to talk about stationary state probabilities: we let μ_i be the stationary state probability of being in state $i \in \mathcal{S}$, i.e., $\mu_i = \Pr(S_\ell = i)$ for all ℓ. (Note that because of an assumption to be made later on, cf. Assumption 34, the FSMS models under consideration will possess a unique stationary state distribution.)
- Let

$$Q_{ij} \triangleq \mu_i \cdot p_{ij} = \Pr(S_{\ell-1} = i, X_\ell = x_{ij}, S_\ell = j) \\ = \Pr(S_{\ell-1} = i, S_\ell = j)$$

  (for all ℓ) be the stationary probability of using the branch with label x_{ij} which goes from state i to state j for $(i,j) \in \mathcal{B}$.
- We let b_ℓ be the branch taken at time ℓ. A legal branch sequence consists of consecutive branches where the ending state of a branch equals the starting state of the next branch; all other branch sequences are called illegal.

Example 16 (Unconstrained Markov Source With Memory Order L): If $\ldots, X_{-1}, X_0, X_1, \ldots$ is a time-invariant Markov process of order L, then $Q(x_\ell | \boldsymbol{x}_{-\infty}^{\ell-1}) = Q(x_\ell | \boldsymbol{x}_{\ell-L}^{\ell-1})$. In order to obtain a source as in Definition 14, we introduce the state sequence $\ldots, S_{-1}, S_0, S_1, \ldots$ with $S_\ell = X_{\ell-L+1}^{\ell}$ for all ℓ. With the notation of Definition 15, this results in a trellis with $|\mathcal{S}| = |\mathcal{X}|^L$ states and $|\mathcal{X}|$ outgoing branches per state.

The trellis in Fig. 5 is actually the trellis representing such a process for $L = 2$ and source alphabet $\mathcal{X} = \{-1,+1\}$. Here state "1" corresponds to $S_\ell = (X_{\ell-1}, X_\ell) = (-1,-1)$, state "2" corresponds to $S_\ell = (X_{\ell-1}, X_\ell) = (-1,+1)$, state "3" corresponds to $S_\ell = (X_{\ell-1}, X_\ell) = (+1,-1)$, and state "4" corresponds to $S_\ell = (X_{\ell-1}, X_\ell) = (+1,+1)$.

The trellis section in Fig. 9 (top), on the other hand, corresponds to a process with $L = 1$ and source alphabet $\mathcal{X} = \{-1,+1\}$.

Example 17 (Run-Length Constrained Markov Source): Consider a Markov process $\ldots, X_{-1}, X_0, X_1, \ldots$ with $\mathcal{X} = \{0,1\}$ where not all subsequences are allowed; (d,k) run-length limited codes are a typical example (see, e.g., [17]). Introducing an appropriate number of states, such processes can also be cast in the framework of Definition 14. Fig. 10 (top) shows, e.g., the resulting trellis section of an $(1,\infty)$-constrained source process, *i.e.*, where the number of zeros between two ones is constrained to be in the range 1 to ∞, so no two consecutive ones can appear in the sequence.

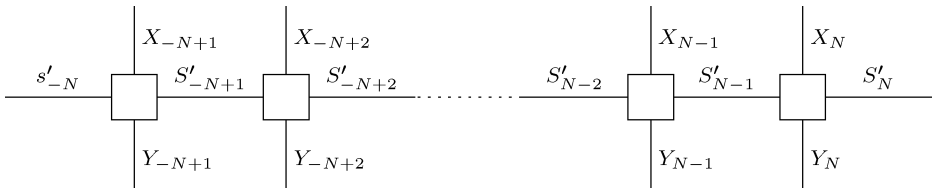

Fig. 6. Normal factor graph representing the joint pmf (conditioned on s'_{-N}) of an FSMC model.

B. FSMCs

Definition 18 (FSMCs): A time-invariant (discrete) FSMC [15] has an input process $\ldots, X_{-1}, X_0, X_1, \ldots$, an output process $\ldots, Y_{-1}, Y_0, Y_1, \ldots$, and a state process $\ldots, S'_{-1}, S'_0, S'_1, \ldots$, where $X_\ell \in \mathcal{X}$, $Y_\ell \in \mathcal{Y}$, and $S'_\ell \in \mathcal{S}'$ for all $\ell \in \mathbb{Z}$. We assume that the alphabets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{S}'$ are finite and that for any $N > 0$ the joint pmf decomposes as

$$P_{\boldsymbol{S}'^N_{-N+1}, \boldsymbol{Y}^N_{-N+1} | S'_{-N}, \boldsymbol{X}^N_{-N+1}}(\boldsymbol{s}'^N_{-N+1}, \boldsymbol{y}^N_{-N+1} | s'_{-N}, \boldsymbol{x}^N_{-N+1}) = \prod_{\ell \in \mathcal{I}_N} P_{S'_\ell, Y_\ell | S'_{\ell-1}, X_\ell}(s'_\ell, y_\ell | s'_{\ell-1}, x_\ell)$$

where $P_{S'_\ell, Y_\ell | S'_{\ell-1}, X_\ell}$ is independent of ℓ. This factorization is shown in the normal factor graph in Fig. 6.We will use the notation

$$W(s'_\ell, y_\ell | s'_{\ell-1}, x_\ell) \triangleq P_{S'_\ell, Y_\ell | S'_{\ell-1}, X_\ell}(s'_\ell, y_\ell | s'_{\ell-1}, x_\ell)$$

$$W(\boldsymbol{s}'^N_{-N+1}, \boldsymbol{y}^N_{-N+1} | s'_{-N}, \boldsymbol{x}^N_{-N+1}) \triangleq P_{\boldsymbol{S}'^N_{-N+1}, \boldsymbol{Y}^N_{-N+1} | S'_{-N}, \boldsymbol{X}^N_{-N+1}}(\boldsymbol{s}'^N_{-N+1}, \boldsymbol{y}^N_{-N+1} | s'_{-N}, \boldsymbol{x}^N_{-N+1}),$$

with the derived quantity

$$W(\boldsymbol{y}^N_{-N+1} | s'_{-N}, \boldsymbol{x}^N_{-N+1}) \triangleq \sum_{\boldsymbol{s}'^N_{-N+1}} W(\boldsymbol{s}'^N_{-N+1}, \boldsymbol{y}^N_{-N+1} | s'_{-N}, \boldsymbol{x}^N_{-N+1}).$$

In this paper, we consider only indecomposable FSMCs as defined by Gallager [15], i.e., channels where roughly speaking the influence of the initial state fades out with time for every possible channel input sequence. (For the exact definition, see [15, Ch. 4.6].)

Again, such a channel with channel law

$$W(s'_\ell, y_\ell | s'_{\ell-1}, x_\ell) = W(s'_\ell | s'_{\ell-1}, x_\ell) \cdot W(y_\ell | s'_{\ell-1}, x_\ell, s'_\ell)$$

can be represented by a (time-invariant) trellis with $|\mathcal{S}'|$ states where the set $\mathcal{B}'$ describes the legal transitions and where for each branch b'_ℓ at time ℓ there is a branch label $(x_\ell; W(s'_\ell | s'_{\ell-1}, x_\ell))$ showing the input symbol and the transition probability. Often, also the "noiseless channel output" is included in the branch label. Note that we allow the trellis representing the FSMC to have parallel branches. (To be precise, though, we can assume without loss of generality that for a given triple $(s'_{\ell-1}, x_\ell, s'_\ell)$ there is at most one branch between the states $s'_{\ell-1}$ and s'_ℓ whose label is x_ℓ.)

In this paper, we discuss only FSMCs for which

$$W(y_\ell | s'_{\ell-1}, x_\ell, s'_\ell) > 0, \quad \text{for all } (s'_{\ell-1}, x_\ell, s'_\ell) \in \mathcal{B}'$$

and for all $y_\ell \in \mathcal{Y}$. For finite N we do not need this technical condition, however, it will be useful in the limit $N \to \infty$ where we can use results in the style of [42]–[44] for interchanging the limit $N \to \infty$ with (the implicit limit when taking) derivatives.

Example 19 (Finite Impulse Response (FIR) Filter With Additive White Noise): Let $\mathcal{X}$ be a finite subset of $\mathbb{R}$. We consider a channel where the output process is given by $Y_\ell = \sum_{k=0}^m h_k X_{\ell-k} + Z_\ell$. Here we assume to have $X_\ell \in \mathcal{X}, h_k \in \mathbb{R}$, and that $\ldots, Z_{-1}, Z_0, Z_1, \ldots$ is a white noise process with $Z_\ell \in \mathbb{R}$. This type of channel is also known as partial response channel with partial response polynomial $H(D) = \sum_{k=0}^m h_k D^k$ and can be represented by a trellis having the state alphabet $\mathcal{S}' = \mathcal{X}^m$ with $|\mathcal{S}'| = |\mathcal{X}|^m$ states. Fig. 9 (middle) shows such a channel with $m = 1$ where $\mathcal{X} = \{-1, +1\}$ and $h_0 = 1$, $h_1 = -1$. (Note that the output alphabet in this example is continuous whereas Definition 18 required the output alphabet to be finite. There are two possible solutions to this: the channel in the example can be approximated to any desired degree by a channel having finite output alphabet or one can show that the results in this paper hold also for continuous output alphabets provided some suitable conditions hold (these conditions are similar to the ones mentioned in [19]).

Example 20 (Gilbert–Elliott Channel): The Gilbert–Elliott channel [15] has the state alphabet $\mathcal{S}' = \{\mathrm{b}, \mathrm{g}\}$, i.e., a "bad" state and a "good" state, the input alphabet $\mathcal{X} = \{0, 1\}$, and the output alphabet $\{0, 1\}$. One defines

$$W(s'_\ell, y_\ell | s'_{\ell-1}, x_\ell) \triangleq W(s'_\ell | s'_{\ell-1}) \cdot W(y_\ell | s'_{\ell-1}, x_\ell)$$

with $W(\mathrm{b}|\mathrm{g}) = p_\mathrm{b}$, $W(\mathrm{g}|\mathrm{b}) = p_\mathrm{g}$, and $W(y_\ell | s'_{\ell-1}, x_\ell)$ is a binary symmetric channel (BSC) with crossover probability ϵ_g when $s'_{\ell-1} = \mathrm{g}$ and a BSC with crossover probability ϵ_b when $s'_{\ell-1} = \mathrm{b}$, respectively. The trellis section of such a channel is shown in Fig. 10 (middle).

Remark 21 (Controllability of the Channel State): There is a fundamental difference between the channels in Example 19 and in Example 20, respectively: whereas in the former case the input can steer the channel state into any desired state (when allowing sufficient time), the input has no influence on the channel state whatsoever in the latter case. Of course, one can come up with various intermediate channel models where the channel state can only partially be controlled.

Definition 22 (Controllable Channel): If an indecomposable FSMC can be taken from any state into any other state by a finite number of channel inputs which do not depend on the current state, the channel is called *controllable*. (Referring to [15, first paragraph on p. 111 and Example 4.26 on p. 527], we note that there are also decomposable channels that could be called controllable and for which the unconstrained capacity is uniquely defined. However, in the following we will not consider such channels because we deal exclusively with indecomposable channels.)

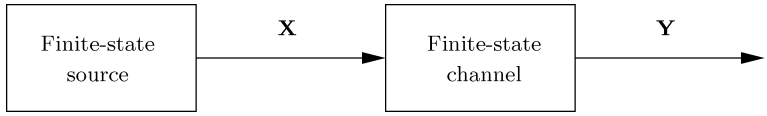

Fig. 7. FSMS and FSMC.

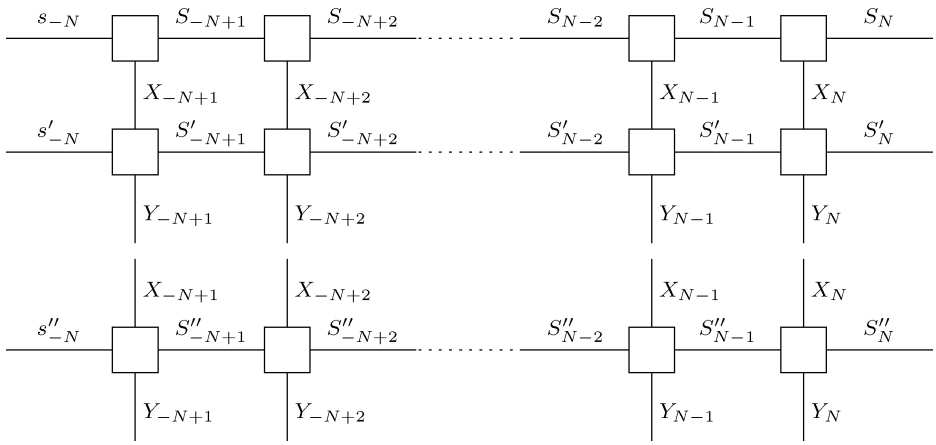

Fig. 8. Top: normal factor graph representing the joint pmf (conditioned on s_{-N} and s'_{-N}) of the concatenation of an FSMS and an FSMC. Bottom: normal factor graph of the same joint pmf but now with the new overall states S''_ℓ, $\ell \in \mathcal{I}_N$.

C. FSMJS/Cs

In this subsection, we combine an FSMS and an FSMC to a finite-state machine joint source/channel (FSMJS/C) model.

Definition 23 (FSMJS/C): Consider an FSMS model as in Definition 14 and an FSMC model as in Definition 18. If the output alphabet of the former equals the input alphabet of the latter we can combine the two models as shown in the block diagram in Fig. 7.

Defining the new state sequence $\ldots, s''_{-1}, s''_0, s''_1, \ldots$, where $s''_\ell \triangleq (s_\ell, s'_\ell) \in \mathcal{S}''$ for all $\ell \in \mathbb{Z}$ and $\mathcal{S}'' \triangleq \mathcal{S} \times \mathcal{S}'$, the joint pmf is then given by

$$\begin{aligned} &P_{\boldsymbol{S}''^N_{-N+1}\boldsymbol{X}^N_{-N+1}\boldsymbol{Y}^N_{-N+1}|S''_{-N}}(\boldsymbol{s}''^N_{-N+1}, \boldsymbol{x}^N_{-N+1}, \boldsymbol{y}^N_{-N+1}|s''_{-N}) \\ &= \prod_{\ell\in\mathcal{I}_N} P_{S_\ell,X_\ell|S_{\ell-1}}(s_\ell, x_\ell|s_{\ell-1}) \\ &\qquad \cdot P_{S'_\ell,Y_\ell|S'_{\ell-1},X_\ell}(s'_\ell, y_\ell|s'_{\ell-1}, x_\ell) \\ &= \prod_{\ell\in\mathcal{I}_N} Q(s_\ell, x_\ell|s_{\ell-1}) W(s'_\ell, y_\ell|s'_{\ell-1}, x_\ell). \end{aligned} \tag{3}$$

This factorization is shown in the normal factor graph in Fig. 8 (top). Again, such an FSMJS/C model can be described by a (time-invariant) trellis with $|\mathcal{S}''|$ states, where the set $\mathcal{B}''$ denotes the legal transitions and where a branch at time index ℓ will be denoted by b''_ℓ.

Remark 24 (Notational Conventions): In (3), we implicitly used the fact that given s''_ℓ, we know s_ℓ and s'_ℓ. We will denote this as $s_\ell = s_\ell(s''_\ell)$ and $s'_\ell = s'_\ell(s''_\ell)$, respectively. A legal branch b''_ℓ determines implicitly b_ℓ and b'_ℓ; we write this as $b_\ell = b_\ell(b''_\ell)$ and $b'_\ell = b'_\ell(b''_\ell)$, respectively. A legal branch sequence $\boldsymbol{b}''$ also determines the state sequences $\boldsymbol{s}$, $\boldsymbol{s}'$, and $\boldsymbol{s}''$ uniquely; we will write this as $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{b}'')$, $\boldsymbol{s}' = \boldsymbol{s}'(\boldsymbol{b}'')$, and $\boldsymbol{s}'' = \boldsymbol{s}''(\boldsymbol{b}'')$. (See also the remarks in Section I-B.)

Note that similarly to the case of the DMC in Section II, the channel law W in Definition 23 is **not** a function of the source law Q. Note also that in some FSMJS/C models not all states of $\mathcal{S}'' = \mathcal{S} \times \mathcal{S}'$ can be reached. In this case, one can reduce the necessary state–space size to describe the FSMJS/C model. This happens, e.g., in Example 25.

Example 25 (FSMJS/C Model 1): Fig. 9 (bottom) shows the overall trellis of a Markov source with memory 1 as in Example 16 and an FIR filter with $m = 1$ and additive white noise as in Example 19.

Example 26 (FSMJS/C Model 2): Fig. 10 (bottom) shows the overall trellis of an $(1, \infty)$-run-length limited (RLL) input process as in Example 17 with a Gilbert–Elliott channel as in Example 20.

The factor graphs of the FSMS (shown in Fig. 4) and the FSMC (shown in Fig. 6) are trees. But note that the factor graph of the FSMJS/C in Fig. 8 (top) is not a tree anymore. To obtain the tree property we have to merge for each time index a pair of function nodes to one function node and replace the pair of edges (S_ℓ, S'_ℓ) by the single edge S''_ℓ as shown in Fig. 8 (bottom). The tree property is crucial for being able to efficiently compute exact pmf marginals using the sum–product algorithm [39]. Moreover, conditioning on a state S''_ℓ makes the past and the future stochastically independent, a property well known from hidden Markov models.

Because we will consider a finite window of the input, output, and state processes, it makes sense to introduce the following notation that will be used throughout the rest of this paper (see also Definition 13).

Definition 27 (Notation): We assume N to be a positive integer; note that in all our results we will mainly be interested in the limit $N \to \infty$.

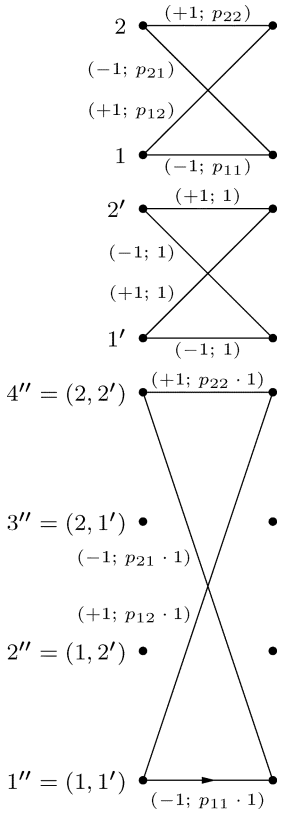

Fig. 9. Trellis sections for Example 25. Top: trellis section of the source model. Middle: trellis section of the channel model. Bottom: trellis section of the FSMJS/C model.

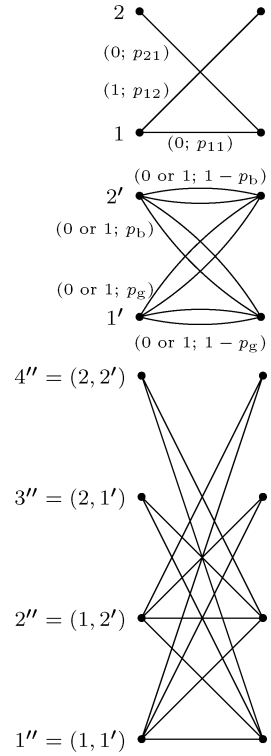

Fig. 10. Trellis sections for Example 26. Top: trellis section of the source model. Middle: trellis section of the channel model. Bottom: trellis section of the FSMJS/C model. (Because of space constraints we omitted the branch labels in the bottom trellis.) Note that whereas the middle trellis contains parallel branches, the bottom trellis does not.

- We define the following finite windows of the input, output, and state processes:

$$\begin{aligned}&\boldsymbol{x} \triangleq \boldsymbol{x}_{-N+1}^{N}, \quad \boldsymbol{y} \triangleq \boldsymbol{y}_{-N+1}^{N},\\ &\boldsymbol{s} \triangleq \boldsymbol{s}_{-N}^{N}, \quad \boldsymbol{s}' \triangleq \boldsymbol{s}_{-N}'^{N}, \quad \boldsymbol{s}'' \triangleq \boldsymbol{s}''^{N}_{-N},\\ &\boldsymbol{b} \triangleq \boldsymbol{b}_{-N+1}^{N}, \quad \boldsymbol{b}' \triangleq \boldsymbol{b}_{-N+1}'^{N}, \quad \boldsymbol{b}'' \triangleq \boldsymbol{b}''^{N}_{-N+1}.\end{aligned}$$

  (Note the slight differences in the lower indices.)
- Usually (and as already mentioned in Section I-B), we will write $\sum_{\boldsymbol{b}}$ to denote the sum over all legal branch sequences. Moreover, in summations like $\sum_{\boldsymbol{b}} Q(\boldsymbol{b}) f(Q(\boldsymbol{b}))$ (where $f : \mathbb{R} \to \mathbb{R}$ is some function, typically the logarithm function) we sum only over legal sequences $\boldsymbol{b}$ where $Q(\boldsymbol{b})$ is nonzero. Similar conventions will be used for other random variables and vectors.

D. Unconstrained FSMC Capacity

This subsection focuses on the unconstrained channel capacity for FSMCs as considered by Gallager [15].

Definition 28 (Set of All Joint Input PMFs): Let $\mathcal{X}$ be the input alphabet of a finite-state channel. For some $N > 0$, let $\mathcal{Q}^{(N)}$ be the set of all pmfs defined over the set $\mathcal{X}_{-N+1}^{N}$, i.e.,

$$\mathcal{Q}^{(N)} \triangleq \left\{ Q : \mathcal{X}_{-N+1}^{N} \to \mathbb{R} \;\middle|\; \begin{array}{l} Q(\boldsymbol{x}) \geq 0 \;\; \forall\, \boldsymbol{x} \in \mathcal{X}_{-N+1}^{N} \\ \sum_{\boldsymbol{x} \in \mathcal{X}_{-N+1}^{N}} Q(\boldsymbol{x}) = 1 \end{array} \right\}.$$

Definition 29 (Mutual Information of an FSMC): Let $\mathcal{X}$ be the input alphabet of a finite-state channel with channel law $W(\boldsymbol{y}|\boldsymbol{x}, s'_{-N})$ as defined in Definition 18. Let $Q \in \mathcal{Q}^{(N)}$. The output pmf is denoted by $(QW)\left(\boldsymbol{y}|s'_{-N}\right)$. The mutual information between the channel input and output (conditioned on $S'_{-N} = s'_{-N}$) is then

$$\begin{aligned}&I^{(N)}(Q, W, s'_{-N})\\ &\quad \triangleq I\left(\boldsymbol{X}; \boldsymbol{Y} \middle| S'_{-N}{=}s'_{-N}\right)\\ &\quad \triangleq \sum_{\boldsymbol{x}} \sum_{\boldsymbol{y}} Q(\boldsymbol{x}) W(\boldsymbol{y}|\boldsymbol{x}, s'_{-N}) \log\left(\frac{W(\boldsymbol{y}|\boldsymbol{x}, s'_{-N})}{(QW)(\boldsymbol{y}|s'_{-N})}\right).\end{aligned}$$

Definition 30 (Unconstrained FSMC Capacity): Let an FSMC with channel law W be given. Its unconstrained FSMC capacity (or simply unconstrained capacity) is defined to be

$$C(W) \triangleq \lim_{N\to\infty} \max_{Q \in \mathcal{Q}^{(N)}} \frac{1}{2N} I^{(N)}(Q, W, s'_{-N}) \tag{4}$$

with $\mathcal{Q}^{(N)}$ and the mutual information as given in Definitions 28 and 29, respectively.

Note that the limit in (4) does not depend on the initial state s'_{-N} because the channel is assumed to be indecomposable [15, Theorem 4.6.4].

E. Constrained FSMC Capacity

We assume to have an FSMJS/C model as given in Definition 23; while the channel part will be assumed to be fixed, we will be allowed to vary the source parameters that are compatible with a given set $\mathcal{B}$. Informally, the *FSMJS/C capacity* is then defined to be the maximal achievable information rate between the channel input and output when optimized over all source

parameters that are compatible with the set $\mathcal{B}$. The following paragraphs will state the problem more precisely.

First, we need a parameterization of a finite-state source process that is convenient for our purposes, i.e., over which we can easily optimize. In the information-rate function and other functions to come, p_{ij}, μ_i, and Q_{ij} (see Definition 15) will all appear very often; in other words, we need a parameterization that leads to formulas that are as simple as possible in these quantities. To select the *base* parameterization, we define different manifolds and study their advantages and disadvantages. (In the following, we assume that the set $\mathcal{S}$ is implicitly given by the set $\mathcal{B}$.)

Definition 31 (FSMS Manifold $\mathcal{P}$): Let $\mathcal{B}$ be a set of legal branches. We define the manifold $\mathcal{P}$ to be

$$\mathcal{P} \triangleq \mathcal{P}(\mathcal{B}) \triangleq \left\{ \{p_{ij}\}_{(i,j)\in\mathcal{B}} \left| \begin{array}{l} p_{ij} \geq 0 \ \ \forall (i,j) \in \mathcal{B} \\ \sum\limits_{j \in \vec{\mathcal{B}}_i} p_{ij} = 1 \ \ \forall i \in \mathcal{S} \end{array} \right. \right\}. \quad (5)$$

Definition 32 (FSMS Manifold $\mathcal{P}'$): Let $\mathcal{B}$ be a set of legal branches. We define the manifold $\mathcal{P}'$ to be

$$\begin{aligned} \mathcal{P}' &\triangleq \mathcal{P}'(\mathcal{B}) \\ &\triangleq \left\{ \left\{ \begin{array}{c} \{p_{ij}\}_{(i,j)\in\mathcal{B}} \\ \{\mu_i\}_{i\in\mathcal{S}} \end{array} \right\} \left| \begin{array}{l} p_{ij} \geq 0 \ \ \forall (i,j) \in \mathcal{B} \\ \sum\limits_{j \in \vec{\mathcal{B}}_i} p_{ij} = 1 \ \ \forall i \in \mathcal{S} \\ \mu_j = \sum\limits_{i \in \overleftarrow{\mathcal{B}}_j} \mu_i p_{ij} \ \ \forall j \in \mathcal{S} \end{array} \right. \right\}. \end{aligned} \quad (6)$$

Definition 33 (FSMS Manifold $\mathcal{Q}$): Let $\mathcal{B}$ be a set of legal branches. We define the manifold $\mathcal{Q}$ to be

$$\begin{aligned} \mathcal{Q} &\triangleq \mathcal{Q}(\mathcal{B}) \\ &\triangleq \left\{ \{Q_{ij}\}_{(i,j)\in\mathcal{B}} \left| \begin{array}{l} Q_{ij} \geq 0 \ \ \forall (i,j) \in \mathcal{B} \\ \sum\limits_{(i,j)\in\mathcal{B}} Q_{ij} = 1 \\ \sum\limits_{k \in \overleftarrow{\mathcal{B}}_i} Q_{ki} = \sum\limits_{j \in \vec{\mathcal{B}}_i} Q_{ij} \ \ \forall i \in \mathcal{S} \end{array} \right. \right\}. \end{aligned} \quad (7)$$

These three manifolds have the following properties.

- The manifold $\mathcal{P}$ is a polytope, but expressing $\{\mu_i\}$ and $\{Q_{ij}\}$ in terms of $\{p_{ij}\}$ only is quite complicated.
- When working with the manifold $\mathcal{P}'$, we can express $\{Q_{ij}\}$ easily in terms of $\{\mu_i\}$ and $\{p_{ij}\}$. However, $\mathcal{P}'$ has the drawback of being a nonconvex set in general.
- The manifold $\mathcal{Q}$ is a polytope and we can express $\{\mu_i\}$ and $\{p_{ij}\}$ easily in terms of $\{Q_{ij}\}$. Namely

$$\mu_i = \sum_{j \in \vec{\mathcal{B}}_i} Q_{ij} = \sum_{k \in \overleftarrow{\mathcal{B}}_i} Q_{ki} \quad (8)$$

$$p_{ij} = \frac{Q_{ij}}{\mu_i} = \frac{Q_{ij}}{\sum_{j' \in \vec{\mathcal{B}}_i} Q_{ij'}}. \quad (9)$$

(We will point out connections of the manifold $\mathcal{Q}$ to the Bethe free energy in Remark 40.)

Because of these properties, we will choose the manifold $\mathcal{Q}$ so that the set $\{Q_{ij}\}$ will be our *base* parameterization of an FSMS process. The other manifolds could also be used, but from our experience, $\mathcal{Q}$ turns out to be the most suitable because it most elegantly leads to our desired results. So, when we are using μ_i and p_{ij}, they will always be functions of $\{Q_{ij}\}$, as given in (8)–(9). Note that we will usually write $\{Q_{ij}\}$ when we are talking about a point in $\mathcal{Q}$, but in order to avoid too many braces we will *not* show the curly braces when $\{Q_{ij}\}$ appears as the argument of a function. For example, we will write $g_1(Q_{ij})$ and not $g_1(\{Q_{ij}\})$. Note also that strictly speaking, $\{Q_{ij}\} \in \mathcal{Q}$ represents a collection of joint probabilities that constitute a point on the manifold $\mathcal{Q}$. However, for notational simplicity, we shall use the symbol $\{Q_{ij}\}$ to represent an FSMS process: the term "FSMS process $\{Q_{ij}\}$" is synonymous to "FSMS process described by the parameter set $\{Q_{ij}\} \in \mathcal{Q}$."

Assumption 34 (Assumption on FSMS Processes): In the remainder of this paper we will only be interested in sets $\mathcal{B}$ where the FSMSs corresponding to $\mathrm{relint}(\mathcal{Q}(\mathcal{B}))$ are ergodic and nonperiodic. By $\mathrm{relint}(\mathcal{Q}(\mathcal{B}))$ we mean the relative interior of the manifold $\mathcal{Q}(\mathcal{B})$.

Definition 35 (Joint PMF): We consider an FSMJS/C as given in Definition 23; the present definition serves to introduce the joint pmf that will be used throughout the rest of the paper. Let $N > 0$. We will consider a window of the FSMJS/C model between state time index $-N$ and state time index N. Let a $\{Q_{ij}\} \in \mathcal{Q}$ be given. The joint pmf over all random variables in this window shall be

$$P_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{S},\boldsymbol{S}'}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s},\boldsymbol{s}') \triangleq Q(\boldsymbol{s},\boldsymbol{x}) \cdot W(\boldsymbol{s}',\boldsymbol{y}|\boldsymbol{x}). \quad (10)$$

The source-model-describing part $Q(\boldsymbol{s},\boldsymbol{x})$ in (10) shall be such that

$$\begin{aligned} Q(s_{-N}) &\triangleq \mu_{s_{-N}}, \quad \text{for all } s_{-N} \in \mathcal{S} \\ Q(\boldsymbol{s},\boldsymbol{x}) &\triangleq Q(s_{-N}) \cdot Q(\boldsymbol{s}_{-N+1}^{N}, \boldsymbol{x}|s_{-N}) \end{aligned}$$

where $\mu_{s_{-N}}$ and $Q(\boldsymbol{s}_{-N+1}^{N}, \boldsymbol{x}|s_{-N})$ are defined through $\{Q_{ij}\}$. On the other hand, we let the channel-describing part $W(\boldsymbol{s}',\boldsymbol{y}|\boldsymbol{x})$ in (10) be defined through

$$W(s'_{-N}) \geq 0, \text{ arbitrary such that } \sum_{s'_{-N}} W(s'_{-N}) = 1$$

$$W(\boldsymbol{s}',\boldsymbol{y}|\boldsymbol{x}) \triangleq W(s'_{-N}) \cdot W(\boldsymbol{s}'^{N}_{-N+1}, \boldsymbol{y}|s'_{-N}, \boldsymbol{x}).$$

From (10) it follows that

$$\begin{aligned} P_{\boldsymbol{X},\boldsymbol{S}}(\boldsymbol{x},\boldsymbol{s}) &= \sum_{\boldsymbol{s}'} \sum_{\boldsymbol{y}} P_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{S},\boldsymbol{S}'}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s},\boldsymbol{s}') \\ &= Q(\boldsymbol{s},\boldsymbol{x}) = Q(s_{-N}) \cdot Q(\boldsymbol{s}_{-N+1}^{N}, \boldsymbol{x}|s_{-N}) \end{aligned}$$

i.e., the marginal pmf describing the source is time-invariant. One of the consequences is, e.g., that $Q(s_\ell) = \mu_{s_\ell}$ for all $\ell \in \{-N, \ldots, N\}$.

Second, the actual choice of $W(s'_{-N})$ does not matter for the information rate value because of the assumed indecomposability of the finite-state channel and because we are only interested in the limit $N \to \infty$. In other words, $W(s'_\ell)$ will depend on ℓ in general; but in the limit $N \to \infty$ this causes no problem. Taking $W(s'_{-N}|s_{-N})$ instead of $W(s'_{-N})$ with the correct values assigned to $W(s'_{-N}|s_{-N})$ would lead to time invariance, but then we would have to deal with the functional dependence of $W(s'_{-N}|s_{-N})$ upon $\{Q_{ij}\}$.

Definition 36 (Notation): We will use the following notation. We make use of the properties of $\boldsymbol{s}$ and $\boldsymbol{x}$ stated at the end of Definition 14: the source branch sequence $\boldsymbol{b}$ uniquely determines the source state sequence $\boldsymbol{s}$, which uniquely determines the pair $(\boldsymbol{s},\boldsymbol{x})$, which uniquely determines the pair $(s_{-N},\boldsymbol{x})$. Therefore, we set $Q(\boldsymbol{b}) \triangleq Q(\boldsymbol{s}) \triangleq Q(\boldsymbol{s},\boldsymbol{x}) \triangleq Q(s_{-N},\boldsymbol{x})$. The probability of a source branch sequence, of an output sequence given a

source branch sequence, of an output sequence, and of a source branch sequence given an output sequence are then for some fixed $\boldsymbol{b}$ and $\boldsymbol{y}$, respectively

$$\begin{aligned} Q(\boldsymbol{b}) &\triangleq \mu_{s_{-N}} \cdot \prod_{\ell \in \mathcal{I}_N} p_{s_{\ell-1},s_\ell} = \frac{\prod_{\ell \in \mathcal{I}_N} Q_{s_{\ell-1},s_\ell}}{\prod_{\ell \in \mathcal{I}'_N} \mu_{s_\ell}} \\ &= \frac{\prod_{\ell \in \mathcal{I}_N} Q_{s_{\ell-1},s_\ell}}{\prod_{\ell \in \mathcal{I}'_N} \sum_{j \in \vec{\mathcal{B}}_{s_\ell}} Q_{s_\ell,j}} \qquad (11) \\ W(\boldsymbol{y}|\boldsymbol{b}) &\triangleq \sum_{\boldsymbol{s}'} W(\boldsymbol{s}', \boldsymbol{y}|\boldsymbol{b}) \\ R(\boldsymbol{y}) &\triangleq (QW)(\boldsymbol{y}) \triangleq \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) W(\boldsymbol{y}|\boldsymbol{b}) \\ V(\boldsymbol{b}|\boldsymbol{y}) &\triangleq \frac{Q(\boldsymbol{b})W(\boldsymbol{y}|\boldsymbol{b})}{R(\boldsymbol{y})} = \frac{Q(\boldsymbol{b})W(\boldsymbol{y}|\boldsymbol{b})}{(QW)(\boldsymbol{y})}. \qquad (12) \end{aligned}$$

Fix some legal branch sequence $\boldsymbol{b}''$ and let $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{b}'')$ and $\boldsymbol{s}' = \boldsymbol{s}'(\boldsymbol{b}'')$ as usual. Based on $W(\boldsymbol{s}', \boldsymbol{y}|\boldsymbol{b})$, we define the pmf $W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b})$ in the obvious way such that for any $\boldsymbol{y}$ we have

$$W(\boldsymbol{y}|\boldsymbol{b}) = \sum_{\boldsymbol{s}'} W(\boldsymbol{s}', \boldsymbol{y}|\boldsymbol{b}) = \sum_{\boldsymbol{b}''} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}).$$

Definition 37 (Information Rate): We assume to have an FSMJS/C model as in Definition 23 where the FSMS and FSMC are described by $\{Q_{ij}\}$ and the channel law W, respectively, and where we define the joint pmf as in Definition 35. The information rate

$$I(Q_{ij}, W) \triangleq \lim_{N \to \infty} I^{(N)}(Q_{ij}, W) \qquad (13)$$

where

$$\begin{aligned} I^{(N)}(Q_{ij}, W) &\triangleq \frac{1}{2N} I(\boldsymbol{B}; \boldsymbol{Y}) \\ &= \frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{y}} W(\boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{V(\boldsymbol{b}|\boldsymbol{y})}{Q(\boldsymbol{b})}\right) \\ &= \frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{y}} W(\boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{W(\boldsymbol{y}|\boldsymbol{b})}{R(\boldsymbol{y})}\right) \qquad (14) \end{aligned}$$

will be the crucial quantity for the rest of this paper.

Lemma 38 (Mutual Information Rate): The definitions in Definition 37 are motivated by the following facts. Under the same assumptions as there, one can show that the mutual information $I(\boldsymbol{X}; \boldsymbol{Y})$ between the input and output fulfills

$$I^{(N)}(Q_{ij}, W) - \frac{1}{2N} \log |\mathcal{S}| \leq \frac{1}{2N} I(\boldsymbol{X}; \boldsymbol{Y}) \leq I^{(N)}(Q_{ij}, W) \qquad (15)$$

$$\lim_{N \to \infty} \frac{1}{2N} I(\boldsymbol{X}; \boldsymbol{Y}) = I(Q_{ij}, W) \qquad (16)$$

where $\mathcal{S}$ is the state space of the FSMS.

Proof: Using the assumptions made in the second half of Definition 14 we have $I(\boldsymbol{B}; \boldsymbol{Y}) = I(S_{-N}, \boldsymbol{X}; \boldsymbol{Y})$. The statement in (15) then stems from the facts that $I(S_{-N}, \boldsymbol{X}; \boldsymbol{Y}) = I(\boldsymbol{X}; \boldsymbol{Y}) + I(S_{-N}; \boldsymbol{Y}|\boldsymbol{X})$ (by the chain rule of mutual information [16]) and that $0 \leq I(S_{-N}; \boldsymbol{Y}|\boldsymbol{X}) \leq \log |\mathcal{S}|$. Finally, (16) is a simple consequence of (15) and the finiteness of $|\mathcal{S}|$. □

Note that in (13) and (14), the argument of the mutual information rates $I(\cdot, W)$ and $I^{(N)}(\cdot, W)$ is Q_{ij}, rather than just Q. This is done to make a clear distinction between Definitions 2 and 37. In Definition 2, the argument Q shows that the input process is memoryless whereas in Definition 37 the argument Q_{ij} shows that the input is an FSMS process.

When talking about maximizing the information rate, the quantities of interest are obviously $\frac{1}{2N} I(\boldsymbol{X}; \boldsymbol{Y})$ and $\lim_{N \to \infty} \frac{1}{2N} I(\boldsymbol{X}; \boldsymbol{Y})$. But it turns out to be somewhat simpler to maximize $I^{(N)}(Q_{ij}, W)$ and $I(Q_{ij}, W)$. There is no big loss in doing so as the difference between $\frac{1}{2N} I(\boldsymbol{X}; \boldsymbol{Y})$ and $I^{(N)}(Q_{ij}, W)$ is proportional to $1/N$ in the limit $N \to \infty$ and so we have equality of $\lim_{N \to \infty} \frac{1}{2N} I(\boldsymbol{X}; \boldsymbol{Y})$ and $I(Q_{ij}, W)$ as seen in Lemma 38. Therefore, in the following we will focus our attention onto $I^{(N)}(Q_{ij}, W)$ and $I(Q_{ij}, W)$.

Definition 39 ($\mathcal{Q}$-Constrained FSMC Capacity): Let $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ be a given FSMS manifold and let W be the channel law of a given FSMC. (We remind the reader of the assumption made in Assumption 34.) The $\mathcal{Q}$-constrained FSMC capacity is then defined to be

$$C(\mathcal{Q}, W) \triangleq \max_{\{Q_{ij}\} \in \mathcal{Q}} I(Q_{ij}, W).$$

Clearly, $C(\mathcal{Q}, W) \leq C(W)$, where the unconstrained capacity $C(W)$ was defined in (4). Usually, the inequality is strict. The idea here is that, by incrementally increasing the order of the input FSMS process, the FSMS channel capacity $C(\mathcal{Q}, W)$ can closely approach the channel capacity $C(W)$. Indeed, Chen and Siegel [33] show that for FSMCs as in Example 19 one can achieve the channel capacity *arbitrarily* closely.

Remark 40 (Connection to Bethe Free Energy): When formulating the Bethe free energy (see, e.g., [45, Sec. 1.7]) one tries to approximate a pmf over many variables by a product involving only single and pairwise marginals. Whereas in general the approximation error is not zero, the approximation is exact in the case of pmfs that can be represented by a cycle-free factor graph. In this light, it is not astonishing that the pmf of a Markov chain can be written as in (11) where we only used single and pairwise marginals.

There is an issue worthwhile pointing out when going back and forth between the manifold of joint pmfs of Markov chains and the manifold $\mathcal{Q}$. Let $\mathcal{M}$ be the manifold of possible pmfs for a time-invariant Markov chain of length $2N+1$ that lives on a certain trellis and let $\mathcal{N}$ be the manifold of all stationary pmfs (not necessarily Markovian) over the same variables as in these Markov chains. Obviously, the manifold $\mathcal{M}$ can be considered as a submanifold of $\mathcal{N}$. Moreover, note that in nontrivial cases, $\mathcal{M}$ will be nonconvex in the sense that the convex combination of two points of $\mathcal{M}$ will not lie on $\mathcal{M}$ in general.

Using this notation, computing the single and pairwise marginals of a pmf can be seen as a surjective *linear* map $f : \mathcal{N} \to \mathcal{Q}$. Moreover, the restriction $f_{\mathcal{M}} : \mathcal{M} \to \mathcal{Q}$ is injective and its image is $\mathcal{Q}$; therefore, the map $f_{\mathcal{M}} : \mathcal{M} \to \mathcal{Q}$ is an invertible map. But note that whereas the map f from $\mathcal{N}$ to $\mathcal{Q}$ is linear, the inverse map $f_{\mathcal{M}}^{-1}$ from $\mathcal{Q}$ to $\mathcal{M}$ is nonlinear. The nonlinearity of the inverse map $f_{\mathcal{M}}^{-1}$ seems at first sight to be a contradiction to the linearity of f, especially in the light that $\mathcal{Q}$ is a convex set. But there is no contradiction: this peculiarity stems from the fact that the linear map from $\mathcal{N}$ to $\mathcal{Q}$ is rank-deficient. The moral is that we can work either with $\mathcal{M}$ or $\mathcal{Q}$ but $\mathcal{Q}$ has the advantage of being a convex set.

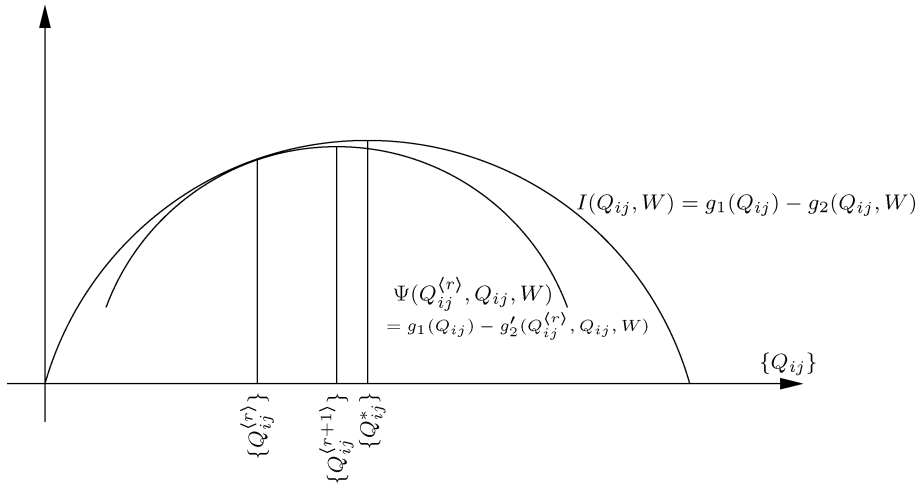

Fig. 11. Generic mutual information rate $I(Q_{ij}, W)$ and surrogate function $\Psi(Q_{ij}^{\langle r\rangle}, Q_{ij}, W)$. The process $\{Q_{ij}^{*}\}$ is a $\mathcal{Q}$-constrained FSMC capacity-achieving input distribution.

IV. The Generalized BAA

This section, together with the next section, is the heart of the paper. We present an algorithm, called the generalized BAA, that optimizes FSMSs at the input of an FSMC in order to optimize the information rate.

Note that the following considerations will all be based on the Definitions 23, 33, 35, 36, 37, 39 and Assumption 34, which were given in the preceding section.

A. The Main Idea Behind the Generalized BAA

From now on, we assume an FSMJS/C model as described in Definition 23. In order to obtain an algorithm (called the generalized BAA) that gives the mutual-information-rate-maximizing parameters of the FSMS, we will extrapolate the classical BAA described in Sections II-C and II-D. That is, we introduce the analogies of the functions T and Ψ for FSMJS/C models. This transition is *not* straightforward; therefore, this subsection is devoted to presenting the main idea of the algorithm. Section IV-B will give the details of the algorithm that should enable a reader to implement it, whereas Section V will give the complete details with the corresponding proofs relegated to the appendices.

Compared to the classical BAA as shown in Section II-C, the generalized BAA for FSMCs works as follows. Again, the algorithm is of an iterative nature. Assume therefore that at iteration r we have found a set $\{Q_{ij}^{\langle r\rangle}\}$ of branch probabilities which lead to an information rate $I(Q_{ij}^{\langle r\rangle}, W)$ (see Fig. 11).[5] At iteration $r+1$ we would like to find a better set $\{Q_{ij}^{\langle r+1\rangle}\}$ of branch probabilities that leads to an information rate with $I(Q_{ij}^{\langle r+1\rangle}, W) \geq I(Q_{ij}^{\langle r\rangle}, W)$ (see Fig. 11). Again, we introduce a surrogate function $\Psi(Q_{ij}^{\langle r\rangle}, Q_{ij}, W)$, which locally (i.e., at $\{Q_{ij}\} = \{Q_{ij}^{\langle r\rangle}\}$) approximates $I(Q_{ij}, W)$ (see Fig. 11). Similar to (2), we write $I(Q_{ij}, W)$ as a difference of two terms

$$I\big(Q_{ij}, W\big) = g_1\big(Q_{ij}\big) - g_2\big(Q_{ij}, W\big) \tag{17}$$

and we let Ψ be the surrogate function

$$\Psi\left(Q_{ij}^{\langle r\rangle}, Q_{ij}, W\right) \triangleq g_1\big(Q_{ij}\big) - g_2'\left(Q_{ij}^{\langle r\rangle}, Q_{ij}, W\right) \tag{18}$$

where $g_2'(Q_{ij}^{\langle r\rangle}, Q_{ij}, W)$ is chosen to be a linear or affine (linear or affine in Q_{ij}) approximation of $g_2(Q_{ij}, W)$ at $\{Q_{ij}\} = \{Q_{ij}^{\langle r\rangle}\}$ (see Fig. 12). The new set $\{Q_{ij}^{\langle r+1\rangle}\}$ of branch probabilities is then chosen to be

$$\left\{Q_{ij}^{\langle r+1\rangle}\right\} \triangleq \arg \max_{\{Q_{ij}\}\in\mathcal{Q}} \Psi\left(Q_{ij}^{\langle r\rangle}, Q_{ij}, W\right).$$

With this approach it follows that stationary points of the algorithm correspond to zero-gradient points of the information rate curve. Moreover, zero-gradient points that are not maxima, are not stable.

Unfortunately, in this case we were neither able to show the concavity of $I(Q_{ij}, W)$, nor the concavity of $g_2(Q_{ij}, W)$ in $\{Q_{ij}\}$. Numerical results suggest the validity of these conjectures, but we still cannot prove or disprove them. (Therefore, we place these problems in Section VII.) The first concavity result would imply that all maxima lie in a connected set,[6] whereas the second concavity result would imply that the algorithm gives at each iteration a new set of branch probabilities whose associated information rate is at least as large as the old information rate. Note that the openness of these conjectures implies that it is not really clear if $I(Q_{ij}, W)$ and $g_2(Q_{ij}, W)$ indeed have the concave shapes that are sketched in Figs. 11 and 12, respectively.

B. Description of the Generalized BAA

Whereas Section IV-A gave just a brief overview of the generalized BAA, this subsection gives the details that enable the reader to apply the algorithm. The complete details can be found in Section V with the corresponding proofs in the appendices.

The next definition introduces the crucial parameters T_{ij} for the generalized BAA; they generalize the parameters $T(x)$ from Section II-D.

[5]Again, the optimization problem is a multidimensional one; but for illustration purposes, a one-dimensional representation of $\{Q_{ij}\}$ will do.

[6]Note that to prove this result, unimodality of $I(Q_{ij}, W)$ in $\{Q_{ij}\}$ would actually be sufficient.

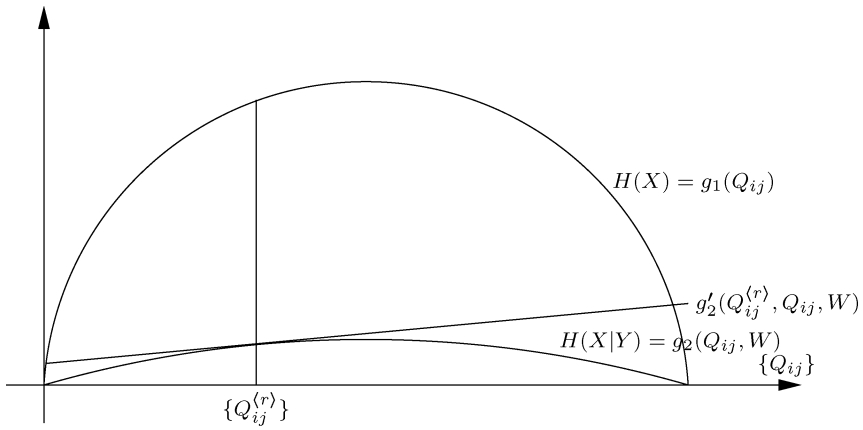

Fig. 12. Generic entropy $H(X) = g_1(Q_{ij})$ and conditional entropy $H(X|Y) = g_2(Q_{ij}, W)$. The function $g_2'(Q_{ij}^{\langle r \rangle}, Q_{ij}, W)$ is a linear approximation of $H(X|Y)$ at $Q_{ij} = Q_{ij}^{\langle r \rangle}$.

Definition 41 (T_{ij} Values): Consider an FSMS process $\{Q_{ij}\} \in \mathcal{Q}(\mathcal{B})$ (from some given manifold $\mathcal{Q}(\mathcal{B})$) and an FSMC with channel law W. For each $(i,j) \in \mathcal{B}$, the $T_{ij}(Q_{ij}, W)$ value[7] is defined to be

$$T_{ij} \triangleq \overline{\overline{T}}_{ij} - \overline{T}_i$$

where

$$\begin{cases} \overline{\overline{T}}_{ij} \triangleq \lim_{N\to\infty} \overline{\overline{T}}_{ij}^{(N)} \\ \overline{T}_i \triangleq \lim_{N\to\infty} \overline{T}_i^{(N)} \end{cases}$$

$$\begin{cases} \overline{\overline{T}}_{ij}^{(N)} \triangleq \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \overline{\overline{T}}_{ij}^{(N)}(\ell) \\ \overline{T}_i^{(N)} \triangleq \frac{1}{2N} \sum_{\ell \in \mathcal{I}'_N} \overline{T}_i^{(N)}(\ell) \end{cases}$$

$$\begin{cases} \overline{\overline{T}}_{ij}^{(N)}(\ell) \triangleq \sum\limits_{\substack{\boldsymbol{b} \\ b_\ell = (i,j)}} Q(\boldsymbol{b}|b_\ell) \sum\limits_{\boldsymbol{b}'} \sum\limits_{\boldsymbol{y}} W(\boldsymbol{b}', \boldsymbol{y}|\boldsymbol{b}) \\ \qquad\qquad \cdot \log\left(\frac{V(b'_\ell|\boldsymbol{y})}{V(b'_\ell|\boldsymbol{b},\boldsymbol{y})} \right) \quad (\text{for } \ell \in \mathcal{I}_N) \\ \overline{T}_i^{(N)}(\ell) \triangleq \sum\limits_{\substack{\boldsymbol{b} \\ s_\ell = i}} Q(\boldsymbol{b}|s_\ell) \sum\limits_{\boldsymbol{b}'} \sum\limits_{\boldsymbol{y}} W(\boldsymbol{b}', \boldsymbol{y}|\boldsymbol{b}) \\ \qquad\qquad \cdot \log\left(\frac{V(s'_\ell|\boldsymbol{y})}{V(s'_\ell|\boldsymbol{b},\boldsymbol{y})} \right) \quad (\text{for } \ell \in \mathcal{I}'_N). \end{cases}$$

Sometimes, we will also use

$$T_{ij}^{(N)} \triangleq \overline{\overline{T}}_{ij}^{(N)} - \overline{T}_i^{(N)}.$$

The following list collects some remarks concerning the T_{ij} values.

- The variable T_{ij} has a *dimension* in the sense that it depends on the choice of the base of the logarithm, therefore expressions like $\mathrm{e}^{T_{ij}}$ have to be modified accordingly if the base of the logarithm is not e.
- Section V-D discusses how the T_{ij} values can be estimated efficiently up to sufficient precision.
- Note that we always normalize by $2N$, despite the fact that $|\mathcal{I}'_N| = 2N - 1$. (In the limit $N \to \infty$ this is irrelevant.)
- Actually, for any T_{ij} defined according to $T_{ij} \triangleq \overline{\overline{T}}_{ij} - \delta\overline{T}_i - (1-\delta)\overline{T}_j$ with $\delta \in \mathbb{R}$, all the upcoming statements can be proven. Using this degree of freedom, the asymmetry in the definition of T_{ij} in Definition 41 can be remedied by choosing $\delta = 1/2$.
- As we will see in Theorem 65, the T_{ij} values can be used, for example, to express the mutual information rate.
- Similar to the comment about $T(x)$ after Definition 4, we can give the following intuition about the quantity T_{ij}: it can be seen as a measure for the "quality of the transition from state i to state j (with symbol x_{ij})" in the following sense. Assume that the transition from state i to state j was used and the symbol x_{ij} was sent and we observe the channel output sequence. Then, the larger T_{ij} is, the larger is the probability of observing a channel output sequence where we can say with high likelihood that the transition from state i to state j was indeed used.
- If an FSMS process with a label different from $\{Q_{ij}\}$ is used, we will decorate the symbol T_{ij}. That is, if the input FSMS is given by $\{\tilde{Q}_{ij}\}$, then we will use $\tilde{T}_{ij} \triangleq T_{ij}(\tilde{Q}_{ij}, W)$.
- Note that in the definition of $\overline{\overline{T}}_{ij}^{(N)}(\ell)$ we could have actually replaced the summation over $\boldsymbol{b}'$ by a summation over b'_ℓ and the term $W(\boldsymbol{b}', \boldsymbol{y}|\boldsymbol{b})$ by the term $W(b'_\ell, \boldsymbol{y}|\boldsymbol{b})$, respectively. Moreover, in the definition of $\overline{T}_i^{(N)}(\ell)$ we could have actually replaced the summation over $\boldsymbol{b}'$ by a summation over s'_ℓ and the term $W(\boldsymbol{b}', \boldsymbol{y}|\boldsymbol{b})$ by the term $W(s'_\ell, \boldsymbol{y}|\boldsymbol{b})$, respectively.
- If the FSMC is controllable, then some simplifications in the definition of $\overline{\overline{T}}_{ij}^{(N)}(\ell)$ and $\overline{T}_i^{(N)}(\ell)$ can be applied. For example, $\boldsymbol{b}$ determines b'_ℓ and s'_ℓ for all $\ell \in \mathcal{I}_N$ (except for ℓ near the boundary of the interval) which means that $V(b'_\ell|\boldsymbol{b},\boldsymbol{y}) = 1$ and $V(s'_\ell|\boldsymbol{b},\boldsymbol{y}) = 1$ for compatible $\boldsymbol{b}$, b'_ℓ, and s'_ℓ.

The next definition generalizes Definition 5.

Definition 42 (Generalized Surrogate Function Ψ): Consider an FSMS process $\{\tilde{Q}_{ij}\} \in \mathrm{relint}(\mathcal{Q}(\mathcal{B}))$ (from some given

[7]We will often not write the arguments (Q_{ij}, W) of the T_{ij} value; sometimes we will use decorations to indicate the arguments.

manifold $\mathcal{Q}(\mathcal{B})$) and an FSMC with channel law W. For any $\{Q_{ij}\} \in \mathcal{Q}(\mathcal{B})$, the generalized function Ψ is defined to be

$$\Psi\left(\tilde{Q}_{ij}, Q_{ij}, W\right) \triangleq \sum_{(i,j)\in\mathcal{B}} Q_{ij} \cdot \left[\log\left(\frac{1}{p_{ij}}\right) + \tilde{T}_{ij}\right].$$

Note that $\{p_{ij}\}$ is implicitly defined in terms of $\{Q_{ij}\}$, cf. (8)–(9), and that the $\tilde{T}_{ij}$'s are calculated according to $\tilde{Q}_{ij}$ and W, i.e., $\tilde{T}_{ij} \triangleq T_{ij}(\tilde{Q}_{ij}, W)$.

Definition 43 (Noisy Adjacency Matrix $\mathbf{A}$): Let $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ be a given FSMS manifold (with state set $\mathcal{S}$) and let W be the channel law of a given FSMC. The noisy adjacency matrix $\mathbf{A} = \mathbf{A}(Q_{ij}, W)$ is a matrix of dimension $|\mathcal{S}| \times |\mathcal{S}|$ with (i,j)th entry

$$\mathsf{A}_{ij} = \begin{cases} \mathrm{e}^{T_{ij}}, & \text{if } (i,j) \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where $T_{ij} \triangleq T_{ij}(Q_{ij}, W)$. If we have an FSMS process $\{\tilde{Q}_{ij}\} \in \mathcal{Q}$ at the channel input, then we denote the corresponding noisy adjacency matrix by $\tilde{\mathbf{A}} \triangleq \mathbf{A}(\tilde{Q}_{ij}, W)$ which is accordingly based on $\tilde{T}_{ij} \triangleq T_{ij}(\tilde{Q}_{ij}, W)$.

When computing the capacity of constrained sequences, such as, for example, RLL sequences, the capacity can be related to the adjacency matrix associated with the corresponding Markov constraint [17]. As we will see, the matrix introduced in the above definition is the natural generalization to the context of noisy channels, therefore it deserves the name *noisy* adjacency matrix.

We next provide the FSMJS/C analogy of Lemma 8.

Lemma 44 (Maximizing Ψ): Let $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ be a given FSMS manifold and let W be the channel law of a given FSMC. Fix an FSMS process $\{\tilde{Q}_{ij}\} \in \mathrm{relint}(\mathcal{Q})$ and define an FSMS process $\{Q^*_{ij}\}$ by

$$\{Q^*_{ij}\} \triangleq \arg \max_{\{Q_{ij}\}\in\mathcal{Q}} \Psi\left(\tilde{Q}_{ij}, Q_{ij}, W\right).$$

$\{Q^*_{ij}\}$ can be determined by the following steps.

- Let $\tilde{\mathbf{A}}$ be the noisy adjacency matrix as defined in Definition 43. (As indicated in Definition 43, $\tilde{\mathbf{A}}$ is the noisy adjacency matrix corresponding to the FSMS $\{\tilde{Q}_{ij}\}$.)
- Let $\tilde{\boldsymbol{\beta}}^{\mathsf{T}}$ and $\tilde{\boldsymbol{\gamma}}$ be the left and right, respectively, eigenvectors corresponding to the maximal (real) eigenvalue $\tilde{\rho}$ of the noisy adjacency matrix $\tilde{\mathbf{A}}$.
- The desired solution is then (using $\tilde{\kappa} \triangleq 1/\sum_{i\in\mathcal{S}} \tilde{\beta}_i\tilde{\gamma}_i$)

$$p^*_{ij} \triangleq \frac{\tilde{\gamma}_j}{\tilde{\gamma}_i} \cdot \frac{\tilde{A}_{ij}}{\tilde{\rho}} \quad (\text{for all } (i,j) \in \mathcal{B}) \tag{20}$$

$$\mu^*_i \triangleq \tilde{\kappa} \cdot \tilde{\beta}_i \cdot \tilde{\gamma}_i \quad (\text{for all } i \in \mathcal{S}) \tag{21}$$

$$Q^*_{ij} \triangleq \mu^*_i \cdot p^*_{ij} \quad (\text{for all } (i,j) \in \mathcal{B}). \tag{22}$$

Note that $\{Q^*_{ij}\}$ fulfills $\{Q^*_{ij}\} \in \mathrm{relint}(\mathcal{Q})$. Moreover, the maximized value of Ψ is

$$\max_{\{Q_{ij}\}\in\mathcal{Q}} \Psi\left(\tilde{Q}_{ij}, Q_{ij}, W\right) = \Psi\left(\tilde{Q}_{ij}, Q^*_{ij}, W\right) = \log(\tilde{\rho}).$$

Proof: See Appendix A. The proof is a somewhat long, but (relatively) straightforward application of the method of Lagrange multipliers. Let us remark that similar problems were solved by Justesen and Høholdt [46] and by Khayrallah and Neuhoff [24]. □

Algorithm 45 (Generalized BAA): Let $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ be a given FSMS manifold and let W be the channel law of a given FSMC. Let $\{Q^{\langle 0\rangle}_{ij}\} \in \mathrm{relint}(\mathcal{Q})$ be some initial (freely chosen) FSMS process. For iterations $r = 0, 1, 2, \ldots$, perform alternatively the following two steps.

- **First Step:** For each $(i,j) \in \mathcal{B}$ calculate $T^{\langle r\rangle}_{ij} \triangleq T_{ij}(Q^{\langle r\rangle}_{ij}, W)$ according to Definition 41. The values $T^{\langle r\rangle}_{ij}$ can be approximated by the procedure given in Section V-C.
- **Second Step:** The new FSMS process $\{Q^{\langle r+1\rangle}_{ij}\}$ is chosen to maximize $\Psi(Q^{\langle r\rangle}_{ij}, Q_{ij}, W)$, i.e.,

$$\left\{Q^{\langle r+1\rangle}_{ij}\right\} \triangleq \arg \max_{\{Q_{ij}\}\in\mathcal{Q}} \Psi\left(Q^{\langle r\rangle}_{ij}, Q_{ij}, W\right)$$

and is calculated according to the algorithm in Lemma 44 with inputs $\{\tilde{Q}_{ij}\} \triangleq \{Q^{\langle r\rangle}_{ij}\}$ and W and output $\{Q^{\langle r+1\rangle}_{ij}\} \triangleq \{Q^*_{ij}\}$.

Theorem 46 (Convergence of the Generalized BAA for FSMCs): Let an FSMC with channel law W be given and let $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ be a manifold of FSMSs consistent with a given set $\mathcal{B}$ of legal branches (see Definition 33). The stationary points of Algorithm 45 correspond to critical points (i.e., local maxima, local minima, and saddle points) of the information rate curve $I(Q_{ij}, W)$ over $\mathcal{Q}$. However, local minima and saddle points cannot be stable stationary points of Algorithm 45. In the case of local maxima, different things can happen.

- If $g_2(Q_{ij}, W)$ is indeed concave in $\{Q_{ij}\}$ (as conjectured in Conjecture 74), then all local maxima of $I(Q_{ij}, W)$ are stable stationary points of Algorithm 45.
- However, if $g_2(Q_{ij}, W)$ turns out to be nonconcave where $I(Q_{ij}, W)$ has a local maximum then it can happen that this local maximum of $I(Q_{ij}, W)$ is not a stable stationary point of Algorithm 45.[8]

Proof: This follows from Theorem 69 in Section V. □

Unfortunately, the only proof of Theorem 46 known to us is very long and requires the introduction of additional notation and several nontrivial intermediate results. This is the reason why we devoted the entire Section V (and the corresponding appendices) to proving Theorem 46 and providing additional insight into the behavior of the generalized BAA. Theorem 46 is the central result of this paper. It justifies the following two statements, which we formulate as corollaries of Theorem 46.

Corollary 47 (Numerically Observed Convergence of the Generalized BAA): If the convergence of the generalized BAA is numerically observed, then the resulting information rate is

[8]In this case, one might want to use a modified (i.e., slowed-down) algorithm that for all $(i,j) \in \mathcal{B}$ replaces $Q^{\langle r+1\rangle}_{ij}$ by $\theta \cdot Q^{\langle r+1\rangle}_{ij} + (1-\theta) \cdot Q^{\langle r\rangle}_{ij}$ after the rth iteration of Algorithm 45. (Here, $0 < \theta \leq 1$ is some suitable constant.) Alternatively, given that the gradient of $I(Q_{ij}, W)$ is easily computed once $\{T_{ij}\}$ is known, one might want to use an off-the-shelf gradient-based search algorithm.

at least a local maximum of the information rate over FSMS processes.

We believe that the generalized BAA actually converges to the *global* maximum, but we were unable to prove it. Nevertheless, all computer simulations we conducted suggest that the algorithm converges to the global maximum. (See also the comments at the end of Section IV-A and in Section VII.) Even though we cannot claim that the global maximum is attained by the algorithm, we can still characterize the $\mathcal{Q}$-constrained FSMC capacity using the next corollary that relates the FSMS channel capacity to the noisy adjacency matrix.

Corollary 48 (Maximal-Eigenvalue Characterization of $\mathcal{Q}$-Constrained FSMC Capacity) : Consider an FSMC with channel law W and let $\{\hat{Q}_{ij}\}$ characterize the FSMS process that achieves the $\mathcal{Q}$-constrained FSMC capacity $C(\mathcal{Q}, w)$. If $\hat{\mathbf{A}}$ denotes the corresponding noisy adjacency matrix and if $\hat{\rho}$ denotes its maximal (real) eigenvalue, then

$$C(\mathcal{Q}, W) = \log(\hat{\rho}).$$

Proof: Follows directly from Lemma 44 and Theorem 46. □

C. Parallel Branches in the FSMS Model

In Definition 14 we required that from state i to state j of an FSMS there exists at most one branch for any $i, j \in \mathcal{S}$. In this subsection (and in Section IV-D), we relax this requirement by allowing parallel branches, but we require that one can infer the branch sequence from the initial state and the input sequence and *vice versa*. For such sources, a generalized BAA can also be derived. The following changes have to be made to Algorithm 45 (proof omitted).

- For each branch b in the FSMS model define $T_b^{\langle r\rangle}$ in the same spirit as $T_{ij}^{\langle r\rangle}$ in Definition 41.
- The modified $T_{ij}^{\langle r\rangle}$ is then $T_{ij}^{\langle r\rangle} \triangleq \log\Big(\sum_b \exp(T_b^{\langle r\rangle})\Big)$, where the summation is over all branches b from state i to state j. (*Attention:* this modified $T_{ij}^{\langle r\rangle}$ can only be used in this algorithm. It cannot be used in the formulas for computing information rates, cf. (41) and (42) in Theorem 65; the information rate formulas have to be modified to use the $T_b^{\langle r\rangle}$ values.)
- Calculate the new state probabilities $\{\mu_i^{\langle r+1\rangle}\}$ and new transition probabilities $\{p_{ij}^{\langle r+1\rangle}\}$ as in Algorithm 45.
- Let b be a branch going from state i to state j. The new transition probability $p_b^{\langle r+1\rangle}$ of choosing branch b when being in state i is

$$p_b^{\langle r+1\rangle} \triangleq p_{ij}^{\langle r+1\rangle} \cdot \frac{\exp\left(T_b^{\langle r\rangle}\right)}{\exp\left(T_{ij}^{\langle r\rangle}\right)}.$$

  The new probability of using branch b from state i to state j is $\mu_i^{\langle r+1\rangle} \cdot p_b^{\langle r+1\rangle}$.

Before we conclude this subsection, let us remark that such a generalization to FSMSs with parallel branches was proposed in [47], [48], although only for controllable FSMCs.

D. The Classical BAA as a Special Case of the Generalized BAA

This subsection aims at showing that the classical BAA is a special case of Algorithm 45. To this end, we will also use the extension shown in Section IV-C.

Consider a memoryless source with n output symbols that produces symbol $b \in \{1, \ldots, n\}$ with probability $Q(b)$. Such a source can be described by an FSMS with one state and with n parallel branches. Similarly, a DMC with n input symbols can be described by an FSMC with one state and n parallel branches.

Assume that we have reached the beginning of iteration r of the generalized BAA. The first step is then to compute $T_b^{\langle r\rangle}$ for each b and $T_{11}^{\langle r\rangle} \triangleq \log\big(\sum_b \exp(T_b)\big)$. The noisy adjacency matrix is then a 1×1-matrix (scalar) $\mathbf{A}^{\langle r\rangle} \triangleq \exp(T_{11}^{\langle r\rangle})$ with the largest eigenvalue trivially found as $\rho^{\langle r\rangle} = \exp(T_{11}^{\langle r\rangle})$ whose associated left and right eigenvectors are scalars $\boldsymbol{\beta}^{\langle r\rangle^{\mathsf{T}}} = 1$ and $\boldsymbol{\gamma}^{\langle r\rangle} = 1$, respectively. With this, we obtain the update rules

$$p_{11}^{\langle r+1\rangle} \triangleq \frac{\gamma_1^{\langle r\rangle}}{\gamma_1^{\langle r\rangle}} \cdot \frac{A_{11}^{\langle r\rangle}}{\rho^{\langle r\rangle}} = \frac{1}{1} \cdot \frac{\exp\left(T_{11}^{\langle r\rangle}\right)}{\exp\left(T_{11}^{\langle r\rangle}\right)} = 1$$

$$\mu_1^{\langle r+1\rangle} \triangleq 1$$

$$p_b^{\langle r+1\rangle} \triangleq p_{11}^{\langle r+1\rangle} \cdot \frac{\exp\left(T_b^{\langle r\rangle}\right)}{\exp\left(T_{11}^{\langle r\rangle}\right)} = \frac{\exp\left(T_b^{\langle r\rangle}\right)}{\sum_{\bar{b}} \exp\left(T_{\bar{b}}^{\langle r\rangle}\right)}$$

$$Q^{\langle r+1\rangle}(b) \triangleq \mu_1^{\langle r+1\rangle} \cdot p_b^{\langle r+1\rangle} = \mu_1^{\langle r+1\rangle} \cdot p_{11}^{\langle r+1\rangle} \cdot \frac{\exp\left(T_b^{\langle r\rangle}\right)}{\exp\left(T_{11}^{\langle r\rangle}\right)}$$

$$= \frac{\exp\left(T_b^{\langle r\rangle}\right)}{\sum_{\bar{b}} \exp\left(T_{\bar{b}}^{\langle r\rangle}\right)}.$$

Observing that the definition of $T_b^{\langle r\rangle}$ collapses to the corresponding definition of $T^{\langle r\rangle}(b)$ for the classical BAA, we conclude that we have just found the update rules of Algorithm 9 (which is the classical BAA). If a capacity-achieving input distribution is reached, the capacity is $C = \log(\rho) = \log\big(\sum_b \exp(T_b)\big)$.

E. Noiseless Channels With Memory as a Special Case of the Generalized BAA

Shannon [9] studied the case of (controllable) noiseless channels with memory. In our setup, the unconstrained FSMC capacity of such channels equals the normalized logarithm of the number of possible paths in the trellis representing the channel.

Because there is no noise in the channel, we can always infer the channel input sequence from the channel output sequence. This implies that $T_{ij} = 0$ if $(i, j) \in \mathcal{B}$. So, in Lemma 44 and Algorithm 45, the matrix $\tilde{\mathbf{A}}$ has an entry $\tilde{\mathrm{A}}_{ij} = 1$ if $(i, j) \in \mathcal{B}$, and $\tilde{\mathrm{A}}_{ij} = 0$ otherwise; i.e., $\tilde{\mathbf{A}}$ is a matrix consisting only of ones and zeros. The resulting matrix is the adjacency matrix of the trellis. (This observation was the motivation to name the matrix $\tilde{\mathbf{A}}$ the noisy adjacency matrix in the case of channels with noise, see the remarks after Definition 43.) It can easily be seen that the generalized BAA converges after one iteration and the capacity is given by the logarithm of the largest (real) eigenvalue of $\tilde{\mathbf{A}}$.

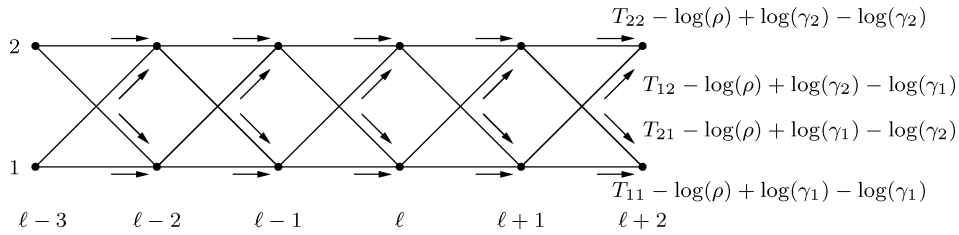

Fig. 13. Trellis of an FSMS with two states. The values next to the arrows indicate the "gain" that one has by choosing a certain direction.

We remark that in the noiseless case a Markov source having the same memory length as the channel is sufficient for achieving the unconstrained FSMC capacity (see also [16, Problem 13 of Ch. 4]). Judging from numerical results, this seems not to be the case in the noisy case: here, the achievable information rates seem to increase as the Markov source memory length increases. But interestingly, when the SNR vanishes then a Markov source having the same memory length as the channel is sufficient for achieving the unconstrained FSMC capacity, see [49]. In conclusion, the case for medium SNR behaves differently than the cases where the SNR goes to $\pm\infty$ decibels.

F. Interpretation of Stationary Points of the Generalized BAA

The main equation in the update formula presented in Algorithm 45 is

$$p_{ij}^{\langle r+1\rangle} = \frac{\gamma_j^{\langle r\rangle}}{\gamma_i^{\langle r\rangle}} \cdot \frac{e^{T_{ij}^{\langle r\rangle}}}{\rho^{\langle r\rangle}} \quad (\text{for all } (i,j) \in \mathcal{B}\,). \tag{23}$$

Assume that we are at a stationary point of the algorithm, i.e., $p_{ij}^{\langle r+1\rangle} = p_{ij}^{\langle r\rangle}$ (for all $(i,j) \in \mathcal{B}$). Setting

$$\begin{aligned} p_{ij} &\triangleq p_{ij}^{\langle r+1\rangle} = p_{ij}^{\langle r\rangle} \quad (\text{for all } (i,j) \in \mathcal{B}\,) \\ T_{ij} &\triangleq T_{ij}^{\langle r+1\rangle} = T_{ij}^{\langle r\rangle} \quad (\text{for all } (i,j) \in \mathcal{B}\,) \\ \gamma_i &\triangleq \gamma_i^{\langle r+1\rangle} = \gamma_i^{\langle r\rangle} \quad (\text{for all } i \in \mathcal{S}\,) \\ \rho &\triangleq \rho^{\langle r+1\rangle} = \rho^{\langle r\rangle} \end{aligned}$$

(23) can be written logarithmically as

$$\log(p_{ij}) = T_{ij} - \log(\rho) + \log(\gamma_j) - \log(\gamma_i).$$

We offer the following interpretation of this formula that characterizes p_{ij} at a stationary point of the generalized BAA.[9] The first term on the right-hand side is T_{ij}: we see that a larger T_{ij} implies a larger p_{ij}. This is in agreement with the intuition given after Definition 41, where we wrote that T_{ij} measures the "quality" of the transition from state i to state j. Fix some time index ℓ. The term $\log(\gamma_j)$ can be seen as measuring the mutual information (+ constant) that can be yielded once we are in state j at time index ℓ, but from this we have to subtract $\log(\gamma_i)$, the mutual information (+ constant) that can be yielded because we were in state i at time index $\ell - 1$. Therefore, the difference $\log(\gamma_j) - \log(\gamma_i)$ measures the advantage/disadvantage to go from state i (at time index $\ell - 1$) to state j (at time index ℓ). (Of course, asymptotically one can transmit the same normalized information starting from any state, but if unnormalized, there is potentially a slight difference.) Finally, the term $-\log(\rho)$ is needed in order to guarantee that $\sum_{i\in\mathcal{S}} p_{ij} = 1$. This interpretation is illustrated in Fig. 13.

Using the left eigenvector of the $\tilde{\mathbf{A}}$ matrix, we equivalently get ($\overleftarrow{p}_{ij}$ is the backward transition probability from state i to state j)

$$\log(\overleftarrow{p}_{ij}) = T_{ji} - \log(\rho) + \log(\beta_j) - \log(\beta_i).$$

We can give a similar interpretation, but now looking to the left (i.e., to the past). For Q_{ij} we get

$$\log(Q_{ij}) = T_{ij} - \log(\rho) + \log(K) + \log(\beta_i) + \log(\gamma_j)$$

where $K = 1/\left(\sum_i \beta_i\gamma_i\right)$. Also, here we can give a similar interpretation, but now looking in both directions (left and right) simultaneously.

Remark 49 (Kuhn–Tucker Conditions): Sometimes not all valid input symbols are used. In the case of a capacity-achieving input pmf Q for a DMC with channel law $W(y|x)$, this is expressed by the Kuhn–Tucker conditions, which state that for any $x \in \mathcal{X}$ we must have

$$\sum_y W(y|x) \log\left(\frac{W(y|x)}{(QW)(y)}\right) \le C$$

with equality if $Q(x) > 0$. For $x \in \mathcal{X}$ with $Q(x) > 0$ this can also be expressed as $-\log(Q(x)) + T(x) = C$.

Similar conditions can be given for the setup in this section. Let $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ be a given FSMS manifold and let W be the channel law of a given FSMC. Furthermore, let $\{Q_{ij}\}$ be a $\mathcal{Q}$-constraint FSMC capacity-achieving input process. Then, there exists constants λ_i, $i \in \mathcal{S}$, such that for any $(i,j) \in \mathcal{B}$ we must have

$$\begin{aligned} \lim_{N\to\infty} \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell=(i,j)}} & Q(\boldsymbol{b}|b_\ell) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}', \boldsymbol{y}|\boldsymbol{b}) \\ & \cdot \log\left(\frac{W(\boldsymbol{y}|b_\ell)}{(QW)(\boldsymbol{y})} \cdot \frac{V(\boldsymbol{b}|\boldsymbol{y}, b_\ell)}{V(\boldsymbol{b}|\boldsymbol{y}, b'_\ell)}\right) \\ - \lim_{N\to\infty} \frac{1}{2N} \sum_{\ell\in\mathcal{I}'_N} \sum_{\substack{\boldsymbol{b} \\ s_\ell=i}} & Q(\boldsymbol{b}|s_\ell) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}', \boldsymbol{y}|\boldsymbol{b}) \\ & \cdot \log\left(\frac{W(\boldsymbol{y}|s_\ell)}{(QW)(\boldsymbol{y})} \cdot \frac{V(\boldsymbol{b}|\boldsymbol{y}, s_\ell)}{V(\boldsymbol{b}|\boldsymbol{y}, s''_\ell)}\right) \\ + \lambda_j - \lambda_i \le C & \end{aligned}$$

with equality if $Q_{ij} > 0$. For $(i,j) \in \mathcal{B}$ with $Q_{ij} > 0$ this condition can be written as $-\log(p_{ij}) + T_{ij} + \lambda_j - \lambda_i = C$. (We omit the proof of these two last statements.)

[9]Note that the right-hand side of this equation depends implicitly on $\{p_{ij}\}$.

TABLE I
SURVEY OVER THE DEFINITIONS AND LEMMAS CONCERNING T_{ij}, $g_1(Q_{ij})$, $g_2(Q_{ij}, W)$, AND $g_2'(\bar{Q}_{ij}, Q_{ij}, W)$

Function definition		Where the function is characterized (lemma and lemma title)		
Def. 41	T_{ij}	Lemma 70	Efficient computation of T_{ij}	
Def. 55	g_1	Lemma 57	Rewriting of g_1	Propty. 1
		Lemma 58	Derivative of g_1	Propty. 2
Def. 55	g_2	Lemma 59	Rewriting of g_2	Propty. 1
		Lemma 60	Derivative of g_2	Propty. 2
		Lemma 63	g_2 vs. g_2'	Propty. 1
		Lemma 64	Deriv. of g_2 vs. deriv. of g_2'	Propty. 2
Def. 62	g_2'	Lemma 63	g_2 vs. g_2'	Propty. 1
		Lemma 64	Deriv. of g_2 vs. deriv. of g_2'	Propty. 2

TABLE II
SURVEY OVER THE DEFINITIONS AND THEOREMS CONCERNING $I(Q_{ij}, W)$ AND $\Psi\left(\bar{Q}_{ij}, Q_{ij}, W\right)$

Function definition		Where the function is characterized (theorem and theorem title)		
Def. 37	I	Th. 65	Rewriting of I	Propty. 1
		Th. 66	Derivative of I	Propty. 2
		Th. 68	I vs. Ψ	Propty. 1
		Th. 69	Deriv. of I vs. deriv. of Ψ	Propty. 2
Def. 67	Ψ	Th. 68	I vs. Ψ	Propty. 1
		Th. 69	Deriv. of I vs. deriv. of Ψ	Propty. 2

V. DETAILS OF THE GENERALIZED BAA

After having given the outline of the generalized BAA in Section IV and before giving some applications of the generalized BAA in Section VI, we proceed now to present all technical details of the algorithm. The main point of departure from the memoryless source and DMC case is that for an FSMJS/C model we need to deal with memory. The influence of memory diminishes exponentially, but nevertheless, at least theoretically it stretches infinitely long. For this reason, we distinguish between two types of functions. 1) Finite-horizon functions (i.e., functions defined for a finite trellis length) have a superscript $^{(N)}$ associated with them. 2) Infinite-horizon functions do not have a superscript because they are obtained by letting the horizon approach infinity, $N \to \infty$.

Throughout this section, we assume to have a given FSMS manifold $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$ (that satisfies Assumption 34) and a given FSMC with channel law W. For any $N > 0$, the joint pmf of an FSMJS/C model will be as stated in Definition 35. (These assumptions will *not* be restated in the body of the following definitions, theorems, and lemmas.) The proofs will be given for finite N only; the expressions for $N \to \infty$ follow by taking the limit in the results for finite N. What quantity is defined where and what properties we prove about them is listed in Tables I and II . Note that all proofs can be found in the appendices.

Assumption 50 (Nonzero-Ness): Note that because of the assumptions at the beginning of the generalized BAA (Algorithm 45) we have $\{Q_{ij}\} \in \operatorname{relint}(\mathcal{Q}(\mathcal{B}))$. For reasons of simplicity, we will show the derivation for the case where $R(\boldsymbol{y}) > 0$ for any $\boldsymbol{y}$ and $V(\boldsymbol{b}''|\boldsymbol{y}) > 0$ for any $\boldsymbol{b}''$ and any $\boldsymbol{y}$. With some more care in the derivations, these conditions can be dropped in the case where N is finite. (Actually, because of the technical condition that we imposed on W in Section III-B, these two conditions are automatically fulfilled.)

In order to understand the generalized BAA, we have to understand how the information rate changes upon varying the FSMS over the FSMS manifold $\mathcal{Q}$. The following approach turns out to be useful.

Definition 51 (Parameterized Family of FSMS Processes): We let $Q_{ij}(\alpha)$ (for each $(i,j) \in \mathcal{B}$) be functions of a real parameter α, where α varies over a suitable range (an α within that range will be called "allowed"). What these functions exactly are is irrelevant, but we assume that they are smooth and that $\{Q_{ij}(\alpha)\} \in \mathcal{Q}$ for any allowed α. The derivative of $Q_{ij}(\alpha)$ with respect to α and evaluated at α' will be denoted by $Q_{ij}^{\alpha}(\alpha')$. Because we defined $\{p_{ij}\}$ and $\{\mu_i\}$ in terms of $\{Q_{ij}\}$ (see comment after Definition 33), we immediately have the functions $\{p_{ij}(\alpha)\}$ and $\{\mu_i(\alpha)\}$ whose derivative with respect to α and evaluated at α' we denote by $\{p_{ij}^{\alpha}(\alpha')\}$ and $\{\mu_i^{\alpha}(\alpha')\}$, respectively. When the evaluation is at $\alpha' = \alpha$, we will often omit the argument (α) and simply write Q_{ij}, Q_{ij}^{α}, p_{ij}, p_{ij}^{α}, μ_i, and μ_i^{α}, instead of $Q_{ij}(\alpha)$, $Q_{ij}^{\alpha}(\alpha)$, $p_{ij}(\alpha)$, $p_{ij}^{\alpha}(\alpha)$, $\mu_i(\alpha)$, and $\mu_i^{\alpha}(\alpha)$, respectively. The following properties follow immediately from (7) and (8)–(9)

$$\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} = 0, \quad \sum_{i\in\mathcal{S}} \mu_i^{\alpha} = 0 \tag{24}$$

$$\sum_{j\in\vec{\mathcal{B}}_i} p_{ij}^{\alpha} = 0 \quad (\text{for all } i \in \mathcal{S}), \tag{25}$$

$$\sum_{k\in\overleftarrow{\mathcal{B}}_i} Q_{ki}^{\alpha} = \sum_{j\in\vec{\mathcal{B}}_i} Q_{ij}^{\alpha} = \mu_i^{\alpha} \quad (\text{for all } i \in \mathcal{S}). \tag{26}$$

(Note that a superscript α will always be related to a derivative and will never be related to raising a certain quantity to the power α.)

A. Some Auxiliary Lemmas

This subsection introduces some lemmas that will be useful when proving the other lemmas and theorems in this section.

Considering the normal factor graph representation in Fig. 8 (bottom) of the joint pmf $Q(\boldsymbol{b})W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b})$, the statements in the following lemma are rather straightforward.

Lemma 52 (Markov Property of a Posteriori PMFs): Using the hidden Markov model structure of the joint pmf $Q(\boldsymbol{b})W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b})$ we have for any $\boldsymbol{b}''$ (and compatible $\boldsymbol{s}''$), any $\boldsymbol{y}$, and any $\ell \in \mathcal{I}_N$

$$V(b_\ell''|\boldsymbol{b}''^{\ell-1}_{-N+1}, \boldsymbol{y}) = V(b_\ell''|s_{\ell-1}'', \boldsymbol{y}) = V(b_\ell''|s_{\ell-1}'', \boldsymbol{y}_\ell^N) \tag{27}$$

$$V(b_\ell''|\boldsymbol{b}''^{N}_{\ell+1}, \boldsymbol{y}) = V(b_\ell''|s_\ell'', \boldsymbol{y}) = V(b_\ell''|s_\ell'', \boldsymbol{y}^{\ell}_{-N+1}). \tag{28}$$

This result says that given $\boldsymbol{y}$, $V(\boldsymbol{b}''|\boldsymbol{y})$ is a Markov probability distribution in $\boldsymbol{b}''$. Using these properties, we can rewrite $V(\boldsymbol{b}''|\boldsymbol{y})$ in two useful different ways as products, namely, for all $\ell \in \mathcal{I}_N$ we have

$$V(\boldsymbol{b}''|\boldsymbol{y}) = V(b_\ell''|\boldsymbol{y}) \cdot V(\boldsymbol{b}''^{N}_{\ell+1}|s_\ell'', \boldsymbol{y}) \cdot V(\boldsymbol{b}''^{\ell-1}_{-N+1}|s_{\ell-1}'', \boldsymbol{y}) \tag{29}$$

$$= V(b_\ell''|\boldsymbol{y}) \cdot V(\boldsymbol{b}''^{N}_{\ell+1}|s_\ell'', \boldsymbol{y}_{\ell+1}^N) \cdot V(\boldsymbol{b}''^{\ell-1}_{-N+1}|s_{\ell-1}'', \boldsymbol{y}^{\ell-1}_{-N+1}) \tag{30}$$

and for all $\ell \in \{-N, \ldots, +N\}$ we have

$$V(\boldsymbol{b}''|\boldsymbol{y}) = V(s''_\ell|\boldsymbol{y}) \cdot V(\boldsymbol{b}''^{N}_{\ell+1}|s''_\ell, \boldsymbol{y}) \cdot V(\boldsymbol{b}''^{\ell}_{-N+1}|s''_\ell, \boldsymbol{y}) \quad (31)$$

$$= V(s''_\ell|\boldsymbol{y}) \cdot V(\boldsymbol{b}''^{N}_{\ell+1}|s''_\ell, \boldsymbol{y}^{N}_{\ell+1}) \cdot V(\boldsymbol{b}''^{\ell}_{-N+1}|s''_\ell, \boldsymbol{y}^{\ell}_{-N+1}). \quad (32)$$

(Note that in (29)–(32) the factorization is done according to "present," "future," and "past.") Similar expressions can also be given for $V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})$.

Proof: All formulas follow from the hidden Markov structure of $Q(\boldsymbol{b}'')W(\boldsymbol{y}|\boldsymbol{b}'')$ and the compatibility of $\boldsymbol{s}''$ and $\boldsymbol{b}''$. □

The next Lemma will be a key part of the main lemma, i.e., Lemma 60.

Lemma 53 (Decomposition of PMF): Consider any $\boldsymbol{b}''$, $\boldsymbol{s}''$, $\boldsymbol{b}$, and $\boldsymbol{y}$, where $\boldsymbol{b}''$, $\boldsymbol{s}''$, and $\boldsymbol{b}$ are assumed to be compatible with each other. Then it holds that

$$V(\boldsymbol{b}|\boldsymbol{y}) = \frac{V(\boldsymbol{b}''|\boldsymbol{y})}{V(\boldsymbol{b}''|\boldsymbol{b}, \boldsymbol{y})}. \quad (33)$$

For all $\ell \in \mathcal{I}_N$ we have

$$V(\boldsymbol{b}|\boldsymbol{y}) = \frac{V(b''_\ell|\boldsymbol{y})}{V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})} \cdot V(\boldsymbol{b}^{N}_{\ell+1}|s''_\ell, \boldsymbol{y}^{N}_{\ell+1}) \cdot V(\boldsymbol{b}^{\ell-1}_{-N+1}|s''_{\ell-1}, \boldsymbol{y}^{\ell-1}_{-N+1}). \quad (34)$$

For all $\ell \in \{-N, \ldots, +N\}$ we have

$$V(\boldsymbol{b}|\boldsymbol{y}) = \frac{V(s''_\ell|\boldsymbol{y})}{V(s''_\ell|\boldsymbol{b}, \boldsymbol{y})} \cdot V(\boldsymbol{b}^{N}_{\ell+1}|s''_\ell, \boldsymbol{y}^{N}_{\ell+1}) \cdot V(\boldsymbol{b}^{\ell}_{-N+1}|s''_\ell, \boldsymbol{y}^{\ell}_{-N+1}). \quad (35)$$

Finally, it holds that

$$V(\boldsymbol{b}''|\boldsymbol{y}) = \left(\prod_{\ell \in \mathcal{I}_N} V(b''_\ell|\boldsymbol{y})\right) \cdot \left(\prod_{\ell \in \mathcal{I}'_N} V(s''_\ell|\boldsymbol{y})\right)^{-1} \quad (36)$$

$$V(\boldsymbol{b}''|\boldsymbol{b}, \boldsymbol{y}) = \left(\prod_{\ell \in \mathcal{I}_N} V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})\right) \cdot \left(\prod_{\ell \in \mathcal{I}'_N} V(s''_\ell|\boldsymbol{b}, \boldsymbol{y})\right)^{-1}. \quad (37)$$

Proof: See Appendix B. □

Note that for a fixed $\boldsymbol{y}$ we can make the same comment about (36) and (37) that we made about (11) in Remark 40.

Lemma 54 (Some Derivatives): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as defined in Definition 51. For any valid branch sequence $\boldsymbol{b}$ we have

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} Q(\boldsymbol{b}) = \left(\sum_{(i,j)\in\mathcal{B}} Q^{\alpha}_{ij} \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \frac{Q(\boldsymbol{b})}{Q_{ij}}\right) - \left(\sum_{(i,j)\in\mathcal{B}} Q^{\alpha}_{ij} \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \frac{Q(\boldsymbol{b})}{\mu_i}\right). \quad (38)$$

Proof: See Appendix B. □

B. Auxiliary and Surrogate Functions and Their Properties

The following definitions and lemmas introduce and characterize the auxiliary and surrogate functions that are used to describe the information rate function, see also Sections IV-A and IV-B.

Definition 55 (Auxiliary Functions g_1 and g_2): We will need the following functions:

$$g_1^{(N)}(Q_{ij}) = -\frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \log(Q(\boldsymbol{b}))$$

$$g_2^{(N)}(Q_{ij}, W) = -\frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{y}} W(\boldsymbol{y}|\boldsymbol{b}) \log\left(V(\boldsymbol{b}|\boldsymbol{y})\right).$$

Additionally, in the limit $N \to \infty$, we will need the functions

$$g_1(Q_{ij}) \triangleq \lim_{N\to\infty} g_1^{(N)}(Q_{ij})$$

$$g_2(Q_{ij}, W) \triangleq \lim_{N\to\infty} g_2^{(N)}(Q_{ij}, W).$$

If $\{Q_{ij}(\alpha)\}$ is a parameterized family of FSMS processes as defined in Definition 51 then we will talk about the functions $g_1^{(N)}(\alpha) \triangleq g_1^{(N)}(Q_{ij}(\alpha))$, $g_2^{(N)}(\alpha, W) \triangleq g_2^{(N)}(Q_{ij}(\alpha), W)$, etc.

Remark 56 (Mutual Information Rate): The mutual information rate functions $I^{(N)}(Q_{ij}, W)$ and $I(Q_{ij}, W)$ were defined in Definition 37. As was alluded to in (17), the functions g_1 and g_2 can be used to express $I^{(N)}(Q_{ij}, W)$ and $I(Q_{ij}, W)$ as a sum of two terms, i.e.,

$$I^{(N)}(Q_{ij}, W) = g_1^{(N)}(Q_{ij}) - g_2^{(N)}(Q_{ij}, W),$$

$$I(Q_{ij}, W) = g_1(Q_{ij}) - g_2(Q_{ij}, W).$$

Lemma 57 (Property 1 of Auxiliary Function g_1): The functions $g_1^{(N)}(Q_{ij})$ and $g_1(Q_{ij})$ as given in Definition 55 can be rewritten as

$$g_1^{(N)}(Q_{ij}) = \sum_{(i,j)\in\mathcal{B}} Q_{ij} \cdot \left[\log\left(\frac{1}{p_{ij}}\right) + \frac{1}{2N}\log\left(\frac{1}{\mu_i}\right)\right]$$

$$g_1(Q_{ij}) = \sum_{(i,j)\in\mathcal{B}} Q_{ij} \log\left(\frac{1}{p_{ij}}\right).$$

Proof: See Appendix B. □

Lemma 58 (Property 2 of Auxiliary Function g_1): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as defined in Definition 51 and let the functions $g_1^{(N)}(\alpha)$ and $g_1(\alpha)$ be given as in Definition 55. Then

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} g_1^{(N)}(\alpha) = \sum_{(i,j)\in\mathcal{B}} Q^{\alpha}_{ij} \cdot \left[\log\left(\frac{1}{p_{ij}}\right) + \frac{1}{2N}\log\left(\frac{1}{\mu_i}\right)\right]$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} g_1(\alpha) = \sum_{(i,j)\in\mathcal{B}} Q^{\alpha}_{ij} \log\left(\frac{1}{p_{ij}}\right).$$

Proof: See Section B. □

Lemma 59 (Property 1 of the Auxiliary Function g_2): The functions $g_2^{(N)}(Q_{ij}, W)$ and $g_2(Q_{ij}, W)$ as given in Definition 55 can be rewritten as

$$\begin{aligned} g_2^{(N)}(Q_{ij}, W) &= -\left[\sum_{(i,j)\in\mathcal{B}} Q_{ij}\overline{\overline{T}}_{ij}^{(N)} - \sum_{i\in\mathcal{S}} \mu_i \overline{T}_i^{(N)}\right] \\ &= -\sum_{(i,j)\in\mathcal{B}} Q_{ij}\cdot T_{ij}^{(N)} \\ g_2(Q_{ij}, W) &= -\left[\sum_{(i,j)\in\mathcal{B}} Q_{ij}\overline{\overline{T}}_{ij} - \sum_{i\in\mathcal{S}} \mu_i \overline{T}_i\right] \\ &= -\sum_{(i,j)\in\mathcal{B}} Q_{ij}\cdot T_{ij} \end{aligned}$$

where $T_{ij}^{(N)} = T_{ij}^{(N)}(Q_{ij}, W)$ and $T_{ij} = T_{ij}(Q_{ij}, W)$ were defined in Definition 41.

Proof: See Appendix B. □

Lemma 60 (Property 2 of the Auxiliary Function g_2): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as defined in Definition 51 and let the functions $g_2^{(N)}(\alpha, W)$ and $g_2(\alpha, W)$ be as given in Definition 55. Then

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} g_2^{(N)}(\alpha, W) = -\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\cdot T_{ij}^{(N)} \tag{39}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} g_2(\alpha, W) = -\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\cdot T_{ij} \tag{40}$$

where $T_{ij}^{(N)} = T_{ij}^{(N)}(Q_{ij}(\alpha), W)$ and $T_{ij} = T_{ij}(Q_{ij}(\alpha), W)$.

Proof: See Appendix B. See also Remark 61. □

Remark 61 (On Property 2 of the Auxiliary Function g_2): This seemingly trivial-looking result in Lemma 60 overcomes the main technical difficulty of this paper; the difficulty lies in the dependency of $T_{ij}^{(N)}$ and T_{ij} on α. Recently, Pfister [51] announced a proof that is quite a bit more compact than the proof in Appendix B, thanks to the use of suitable matrix notation. However, the main steps of Pfister's proof are very similar to our proof.

Definition 62 (Auxiliary Function g_2'): Let $\{Q_{ij}\}, \{\tilde{Q}_{ij}\} \in \mathcal{Q}$. We define the surrogate functions

$$\begin{aligned} {g_2'}^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W) &\triangleq -\sum_{(i,j)\in\mathcal{B}} Q_{ij}\cdot \tilde{T}_{ij}^{(N)} \\ g_2'(\tilde{Q}_{ij}, Q_{ij}, W) &\triangleq -\sum_{(i,j)\in\mathcal{B}} Q_{ij}\cdot \tilde{T}_{ij} \end{aligned}$$

where $\tilde{T}_{ij}^{(N)} = T_{ij}^{(N)}(\tilde{Q}_{ij}, W)$ and $\tilde{T}_{ij} = T_{ij}(\tilde{Q}_{ij}, W)$. If $\{Q_{ij}(\alpha)\}$ is a parameterized family of FSMS processes as introduced in Definition 51 then we define for any allowed scalar $\tilde{\alpha}$

$$\begin{aligned} {g_2'}^{(N)}(\tilde{\alpha}, \alpha, W) &\triangleq {g_2'}^{(N)}(Q_{ij}(\tilde{\alpha}), Q_{ij}(\alpha), W) \\ g_2'(\tilde{\alpha}, \alpha, W) &\triangleq g_2'(Q_{ij}(\tilde{\alpha}), Q_{ij}(\alpha), W). \end{aligned}$$

Lemma 63 (Property 1 of the Auxiliary Function g_2'): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as introduced in Definition 51 and let the functions ${g_2'}^{(N)}(\tilde{\alpha}, \alpha, W)$ and $g_2'(\tilde{\alpha}, \alpha, W)$ be as given in Definition 62. Then for any allowed $\tilde{\alpha}$

$$\begin{aligned} {g_2'}^{(N)}(\tilde{\alpha}, \tilde{\alpha}, W) &= g_2^{(N)}(\tilde{\alpha}, W) \\ g_2'(\tilde{\alpha}, \tilde{\alpha}, W) &= g_2(\tilde{\alpha}, W). \end{aligned}$$

Proof: This follows easily from Definition 62 and Lemma 59. □

Lemma 64 (Property 2 of the Auxiliary Function g_2'): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as introduced in Definition 51 and let the functions ${g_2'}^{(N)}(\tilde{\alpha}, \alpha, W)$ and $g_2'(\tilde{\alpha}, \alpha, W)$ be as given in Definition 62. Then for any allowed $\tilde{\alpha}$

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha} {g_2'}^{(N)}(\tilde{\alpha}, \alpha, W)\Big|_{\alpha=\tilde{\alpha}} &= \frac{\mathrm{d}}{\mathrm{d}\alpha} g_2^{(N)}(\alpha, W)\Big|_{\alpha=\tilde{\alpha}} \\ \frac{\mathrm{d}}{\mathrm{d}\alpha} g_2'(\tilde{\alpha}, \alpha, W)\Big|_{\alpha=\tilde{\alpha}} &= \frac{\mathrm{d}}{\mathrm{d}\alpha} g_2(\alpha, W)\big|_{\alpha=\tilde{\alpha}}. \end{aligned}$$

Proof: See Appendix B. □

Lemmas 63 and 64 show that ${g_2'}^{(N)}(\tilde{\alpha}, \alpha, W)$ is a linear approximation of $g_2^{(N)}(\alpha, W)$ at $\alpha = \tilde{\alpha}$, i.e., at this point they have the same value and the same gradient. The same comment applies to the infinite horizon functions $g_2'(\tilde{\alpha}, \alpha, W)$ and $g_2(\alpha, W)$.

Theorem 65 (Property 1 of the Mutual Information Rate): For any $\{Q_{ij}\} \in \mathcal{Q}$ we have

$$\begin{aligned} &I^{(N)}(Q_{ij}, W) \\ &\quad = \sum_{(i,j)\in\mathcal{B}} Q_{ij}\cdot\left[\log\left(\frac{1}{p_{ij}}\right) + \frac{1}{2N}\log\left(\frac{1}{\mu_i}\right) + T_{ij}^{(N)}\right] \end{aligned} \tag{41}$$

$$I(Q_{ij}, W) = \sum_{(i,j)\in\mathcal{B}} Q_{ij}\cdot\left[\log\left(\frac{1}{p_{ij}}\right) + T_{ij}\right] \tag{42}$$

where $T_{ij}^{(N)} = T_{ij}^{(N)}(Q_{ij}, W)$ and $T_{ij} = T_{ij}(Q_{ij}, W)$. If $\{Q_{ij}(\alpha)\}$ is a parameterized family of FSMS processes as introduced in Definition 51, then we introduce the functions $I^{(N)}(\alpha, W) \triangleq I^{(N)}(Q_{ij}(\alpha), W)$ and $I(\alpha, W) \triangleq I(Q_{ij}(\alpha), W)$.

Proof: See Appendix B. □

Theorem 66 (Property 2 of the Mutual Information Rate): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as introduced in Definition 51. The derivative of $I^{(N)}(\alpha, W)$ and $I(\alpha, W)$ with respect to α is respectively

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\alpha} I^{(N)}(\alpha, W) \\ &\quad = \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\cdot\left[\log\left(\frac{1}{p_{ij}}\right) + \frac{1}{2N}\log\left(\frac{1}{\mu_i}\right) + T_{ij}^{(N)}\right] \\ &\frac{\mathrm{d}}{\mathrm{d}\alpha} I(\alpha, W) \\ &\quad = \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\cdot\left[\log\left(\frac{1}{p_{ij}}\right) + T_{ij}\right] \end{aligned}$$

where $T_{ij}^{(N)} = T_{ij}^{(N)}(Q_{ij}(\alpha), W)$ and $T_{ij} = T_{ij}(Q_{ij}(\alpha), W)$.

Proof: See Appendix B. □

Now we define the (generalized) surrogate function Ψ which is key for the generalized BAA, see also the comments in Sections IV-A and IV-B.

Definition 67 (Generalized Surrogate Function Ψ): We define (see also (18))

$$\Psi^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W) \triangleq g_1^{(N)}(Q_{ij}) - g_2'^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W) \quad (43)$$

$$\Psi(\tilde{Q}_{ij}, Q_{ij}, W) \triangleq g_1(Q_{ij}) - g_2'(\tilde{Q}_{ij}, Q_{ij}, W). \quad (44)$$

Using Lemma 57 and Definition 62, (43)–(44) can be reformulated to read

$$\Psi^{(N)}\big(\tilde{Q}_{ij}, Q_{ij}, W\big) = \sum_{(i,j)\in\mathcal{B}} Q_{ij} \cdot \left[\log\left(\frac{1}{p_{ij}}\right) + \frac{1}{2N}\log\left(\frac{1}{\mu_i}\right) + \tilde{T}_{ij}^{(N)}\right]$$

$$\Psi\big(\tilde{Q}_{ij}, Q_{ij}, W\big) = \sum_{(i,j)\in\mathcal{B}} Q_{ij} \cdot \left[\log\left(\frac{1}{p_{ij}}\right) + \tilde{T}_{ij}\right].$$

Note that the $\tilde{T}_{ij}^{(N)}$'s and $\tilde{T}_{ij}$'s are calculated according to $\tilde{Q}_{ij}$ and W, i.e., $\tilde{T}_{ij}^{(N)} = T_{ij}^{(N)}(\tilde{Q}_{ij}, W)$ and $\tilde{T}_{ij} = T_{ij}(\tilde{Q}_{ij}, W)$. When $\{Q_{ij}(\alpha)\}$ is a parameterized family of FSMS processes as defined in Definition 51 then we introduce the functions $\Psi^{(N)}(\tilde{\alpha}, \alpha, W) \triangleq \Psi^{(N)}(Q_{ij}(\tilde{\alpha}), Q_{ij}(\alpha), W)$ and $\Psi\big(\tilde{\alpha}, \alpha, W\big) \triangleq \Psi(Q_{ij}(\tilde{\alpha}), Q_{ij}(\alpha), W)$.

Theorem 68 (Property 1 of the Generalized Surrogate Function Ψ): For any $\{\tilde{Q}_{ij}\} \in \mathcal{Q}$ we have

$$\Psi^{(N)}\left(\tilde{Q}_{ij}, \tilde{Q}_{ij}, W\right) = I^{(N)}\left(\tilde{Q}_{ij}, W\right)$$

$$\Psi\left(\tilde{Q}_{ij}, \tilde{Q}_{ij}, W\right) = I\left(\tilde{Q}_{ij}, W\right).$$

Proof: In Definition 67 we defined

$$\Psi^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W) \triangleq g_1^{(N)}(Q_{ij}) - g_2'^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W)$$

whereas in Remark 56 we had

$$I^{(N)}(Q_{ij}, W) = g_1^{(N)}(Q_{ij}) - g_2^{(N)}(Q_{ij}, W).$$

Using the relation $g_2'^{(N)}(\tilde{Q}_{ij}, \tilde{Q}_{ij}, W) = g_2^{(N)}(\tilde{Q}_{ij}, W)$, which was established in Lemma 63, we get the desired result. □

Theorem 69 (Property 2 of the Generalized Surrogate Function Ψ): Let $\{Q_{ij}(\alpha)\}$ be a parameterized family of FSMS processes as defined in Definition 51. Then for any allowed $\tilde{\alpha}$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\Psi^{(N)}\left(\tilde{\alpha}, \alpha, W\right)\Big|_{\alpha=\tilde{\alpha}} = \frac{\mathrm{d}}{\mathrm{d}\alpha} I^{(N)}(\alpha, W)\Big|_{\alpha=\tilde{\alpha}}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\Psi\left(\tilde{\alpha}, \alpha, W\right)\Big|_{\alpha=\tilde{\alpha}} = \frac{\mathrm{d}}{\mathrm{d}\alpha} I(\alpha, W)\Big|_{\alpha=\tilde{\alpha}}.$$

Proof: In Definition 67 we defined

$$\Psi^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W) \triangleq g_1^{(N)}(Q_{ij}) - g_2'^{(N)}(\tilde{Q}_{ij}, Q_{ij}, W)$$

whereas in Remark 56 we had

$$I^{(N)}(Q_{ij}, W) = g_1^{(N)}(Q_{ij}) - g_2^{(N)}(Q_{ij}, W).$$

Using Lemma 64 we get the desired result. □

Theorem 69 easily leads to the proof of Theorem 46 in Section IV, which is the main technical result of this paper.

C. Efficient Computation of T_{ij}

Lemma 70 (Efficient Computation of T_{ij}): Consider an FSMS process characterized by $\{Q_{ij}\} \in \mathrm{relint}(\mathcal{Q})$ (for some FSMS manifold $\mathcal{Q} = \mathcal{Q}(\mathcal{B})$) and an FSMC with channel law W. Although the definition of the T_{ij} values is somewhat complicated, these values can indeed be computed quite efficiently. One can use the following steps to get the T_{ij} values with probability one as $N \to \infty$.

- Choose a large N.
- Randomly generate an input sequence and therefore a branch sequence $\check{\boldsymbol{b}}$ (and therefore also the state sequence $\check{\boldsymbol{s}} = \check{\boldsymbol{s}}(\check{\boldsymbol{b}})$). Randomly generate a branch sequence $\check{\boldsymbol{b}}'$ (and therefore also $\check{\boldsymbol{b}}'' = (\check{\boldsymbol{b}}, \check{\boldsymbol{b}}')$) and an output sequence $\check{\boldsymbol{y}}$. (With probability 1 as $N \to \infty$ each of these sequences are typical and together they are jointly typical.)
- Using the BCJR (or forward–backward) algorithm [52], compute for all $b_\ell'' \in \mathcal{B}''$, $\ell \in \mathcal{I}_N$ the quantities $V(b_\ell''|\check{\boldsymbol{y}})$ and $V(b_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})$, and for all $s_\ell'' \in \mathcal{S}''$, $\ell \in \mathcal{I}_N'$ the quantities $V(s_\ell''|\check{\boldsymbol{y}})$ and $V(s_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})$.
- (First possibility) Compute

$$\check{T}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{b}}'', \check{\boldsymbol{y}}) = \check{\overline{\overline{T}}}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{b}}'', \check{\boldsymbol{y}}) - \check{\overline{T}}_i^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{b}}'', \check{\boldsymbol{y}})$$

where

$$\check{\overline{\overline{T}}}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{b}}'', \check{\boldsymbol{y}}) = \frac{1}{2NQ_{ij}} \sum_{\substack{\ell\in\mathcal{I}_N \\ \check{b}_\ell=(i,j)}} \log\left(\frac{V(\check{b}_\ell''|\check{\boldsymbol{y}})}{V(\check{b}_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})}\right)$$

$$\check{\overline{T}}_i^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{b}}'', \check{\boldsymbol{y}}) = \frac{1}{2N\mu_i} \sum_{\substack{\ell\in\mathcal{I}_N' \\ \check{s}_\ell=i}} \log\left(\frac{V(\check{s}_\ell''|\check{\boldsymbol{y}})}{V(\check{s}_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})}\right).$$

- (Second possibility) Computationally better (i.e., better accuracy for smaller N for the cases where some p_{ij}'s are low, i.e., the corresponding branches are visited rarely) are the computation rules

$$\check{T}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}}) = \check{\overline{\overline{T}}}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}}) - \check{\overline{T}}_i^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}})$$

where

$$\check{\overline{\overline{T}}}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}}) = \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N} \sum_{\substack{b_\ell'' \\ b_\ell=(i,j)}} \frac{V(b_\ell''|\check{\boldsymbol{y}})}{Q_{ij}} \log\left(V(b_\ell''|\check{\boldsymbol{y}})\right) - \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N} \sum_{\substack{b_\ell'' \\ b_\ell=\check{b}_\ell=(i,j)}} \frac{V(b_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})}{Q_{ij}} \log\left(V(b_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})\right)$$

$$\check{\overline{T}}_i^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}}) = \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N'} \sum_{\substack{s_\ell'' \\ s_\ell=i}} \frac{V(s_\ell''|\check{\boldsymbol{y}})}{\mu_i} \log\left(V(s_\ell''|\check{\boldsymbol{y}})\right) - \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N} \sum_{\substack{s_\ell'' \\ s_\ell=\check{s}_\ell=i}} \frac{V(s_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})}{\mu_i} \log\left(V(s_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})\right).$$

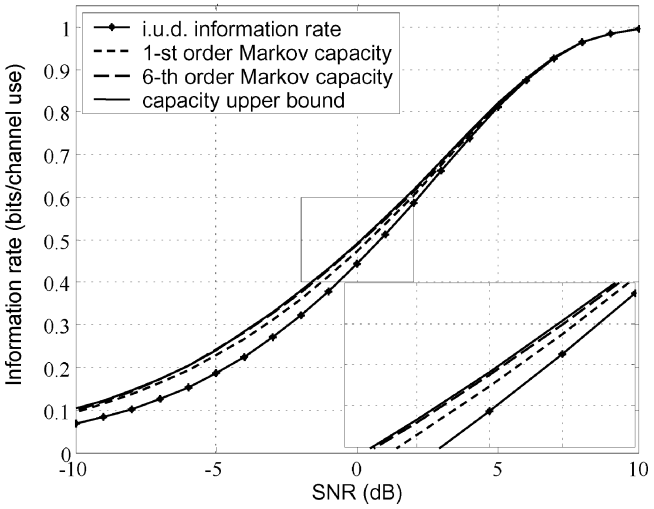

Fig. 14. The i.u.d. information rate, Markov capacities (lower bounds), and upper bound on the capacity of the dicode channel.

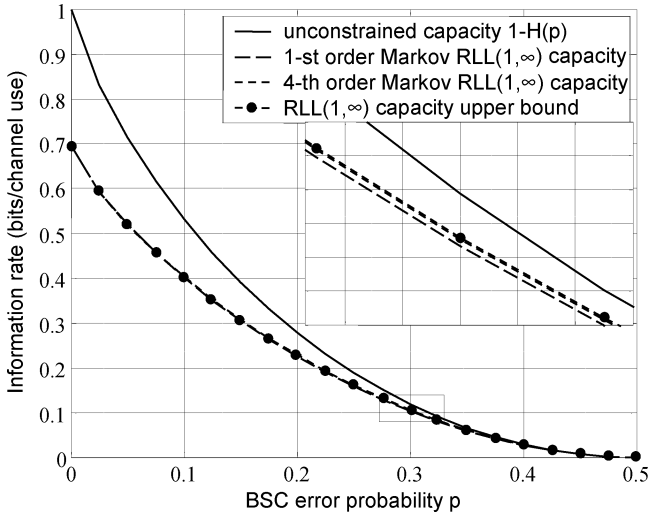

Fig. 15. The unconstrained capacity $1 - H(p)$, Markov capacities (lower bounds), and upper bound on the capacity of $\mathrm{RLL}(1, \infty)$ sequences over the BSC.

This second possibility is close in spirit to the approach taken in [53] to modify the usual procedure to get estimates of bit-error rates.

- $\check{T}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{b}}'', \check{\boldsymbol{y}})$ and $\check{T}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}})$ are estimates of $T_{ij}^{(N)}$ and, as $N \to \infty$, these estimates are equal (with probability 1) to the desired T_{ij}. (We omit the proof of this last statement.)

Proof: See Appendix B. □

VI. Channel Capacity Curves

In this section, we demonstrate the applicability of the generalized BAA by using it to compute (more precisely, to lower-bound) the unconstrained capacities of FSMCs. We will show that the bounds are numerically tight by comparing them to equally tight upper bounds. However, the techniques used to find the numeric upper bounds are beyond the scope of this paper, and can be found in [31], [32].

Example 71 (The Capacity of the Dicode Channel): Consider the dicode channel whose trellis is depicted in Fig. 9 (middle). The channel law is represented by the following equation:

$$Y_\ell = X_\ell - X_{\ell-1} + Z_\ell$$

where Y_ℓ is the channel output at time ℓ, the symbol $X_\ell \in \{-1, +1\}$ stands for the channel input at time ℓ, and Z_ℓ is white Gaussian noise whose variance is σ^2. This channel is known as the *dicode* partial response channel, and its partial response polynomial is $H(D) = 1 - D$. The SNR for this channel (in decibels) is defined as $\mathrm{SNR} = 10 \log_{10} \frac{2}{\sigma^2}$.

To this channel we first apply a sequence of i.u.d. binary random variables, and compute the i.u.d. information rate using the algorithm proposed by Arnold and Loeliger [26], by Sharma and Singh [27], and by Pfister *et al.* [28]. (The algorithm is also given in [19].) The i.u.d. information rate is plotted in Fig. 14. We next apply Markov sources of different orders to the channel inputs, and apply the generalized BAA. Fig. 14 shows the plots for the optimized information rate of a first- and sixth-order Markov input process. To support the claim that the information rate (obtained by running the generalized BAA for a sixth-order Markov process) is a tight lower bound on the channel capacity, we also plot an upper bound. This upper bound is obtained by computing the delayed feedback capacity, which is beyond the scope of this paper, but can be found in [32]. Clearly, the bounds are very close, confirming that the generalized BAA can be used to very accurately estimate the unconstrained channel capacity.

Example 72 (The Capacity of RLL Sequences Over BSCs): An RLL sequence with parameters $(d, k) = (1, \infty)$ is a sequence of 1's and 0's, such that no two consecutive 1's can appear in the sequence, see Example 17. We are going to consider sources that are RLL-$(1, \infty)$ sequences. Such sources can be described by Markov sources, e.g., the trellis section of the corresponding first- order Markov source is shown in Fig. 10 (top). We apply the RLL-$(1, \infty)$ sources to a BSC with parameter p, and ask what is the maximal information rate that can be achieved over the BSC with such a source.

As a reminder, the BSC is a memoryless channel that randomly flips a 0 to a 1 (likewise, flips a 1 to a 0) with probability p. The capacity of the BSC (when no RLL constraint is imposed on the channel input) has been computed by Shannon [9] to be $C = 1 - H(p)$ [bits/channel use], where $H(p)$ is the binary entropy function $H(p) = -p \log p - (1-p) \log(1-p)$. For a reference, the (unconstrained) capacity $1 - H(p)$ [bits/channel use] is plotted in Fig. 15 as a function of p (where $0 \leq p \leq 0.5$).

We applied the generalized BAA to the joint RLL-$(1, \infty)$-source/BSC-channel trellis for the first- and fourth-order Markov sources. The obtained lower bounds on the capacity are plotted in Fig. 15. For a reference, we also plot an upper bound computed by the method in [32]. We see that, indeed, the generalized BAA delivers a numerically very tight lower bound on the capacity, and can readily be used to estimate the capacity of the RLL constraint over a binary symmetric channel.

Example 73 (The Capacity of RLL Sequences Over Gilbert–Elliott Channels): The previous two examples concerned the computation of capacities of two *controllable* FSMCs. We now demonstrate that the generalized BAA applies also to *noncontrollable* FSMCs. We pick the FSMC model that is described in Example 26 and depicted in Fig. 10. The source is constrained

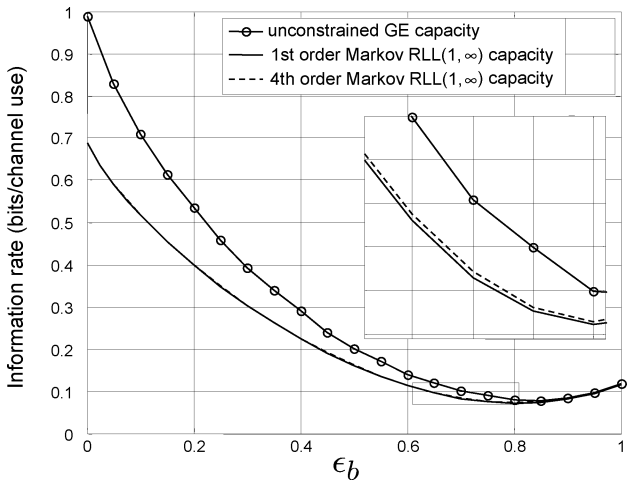

Fig. 16. The unconstrained capacity of a Gilbert–Elliott channel, and lower bounds (Markov capacities) on the capacity of $\mathrm{RLL}(1,\infty)$ sequences over the Gilbert–Elliott channel. The channel parameters are $p_g = P_b = 0.3$, $\epsilon_{\mathrm{g}} = 10^{-3}$, and ϵ_{b} is a varying parameter.

to be an RLL sequence with parameters $(d,k) = (1,\infty)$. The channel is a Gilbert–Elliott channel (see Example 20). The state transition probabilities of the Gilbert–Elliott channel are chosen as $p_{\mathrm{g}} = p_{\mathrm{b}} = 0.3$. The crossover probability of the BSC in the "good" state is $\epsilon_{\mathrm{g}} = 10^{-3}$ while the crossover probability ϵ_{b} of the BSC in the "bad" state is a parameter that varies from 10^{-3} to 1.

Fig. 16 shows the information rates of $\mathrm{RLL}(1,\infty)$ sequences over the Gilbert–Elliott channels as a function of the parameter ϵ_{b}. The curves represent lower bounds on the capacities. We believe that the lower bounds are numerically tight, but, in this case, we cannot support this claim with an equally tight numeric upper bound plot. In general, good numeric/analytic upper bounds on the capacities of uncontrollable FSMCs are unknown (and are open for research[10]). For comparison, in Fig. 16, we also plot the (non-RLL-constrained) capacity of the Gilbert–Elliott channel, which was computed by the Arnold–Loeliger method [26] (see also Pfister, Soriaga, and Siegel [28], and Sharma and Singh [27]), given the knowledge that the capacity is achieved by a Bernoulli-$1/2$ random process [4].

VII. Open Problems

The *global* convergence proof of the classical BAA has two main ingredients: the concavity of $I(Q,W) = g_1(Q) - g_2(Q,W)$ as a function of Q and the concavity of $H(X|Y) = g_2(Q,W)$ as a function of Q. In the case of the generalized BAA, the concavity of the generalizations of these functions is an open issue, see Conjecture 74. (Perhaps the concavity of the $g_1(Q_{ij})$ might help in proving these statements, therefore, we state this known concavity result as Lemma 75.)

Conjecture 74 (Concavity of $g_2(Q_{ij},W)$ and $I(Q_{ij},W)$): We conjecture that $g_2(Q_{ij},W)$ and $I(Q_{ij},W)$ are concave in $\{Q_{ij}\}$.

[10]An approach which has not been pursued yet would be to generalize the results in [31] and see how tight the obtained upper bounds are compared with the known lower bounds.

Lemma 75 (Concavity of $g_1^{(N)}(Q_{ij})$ and $g_1(Q_{ij})$): The functions $g_1^{(N)}(Q_{ij})$ (for any $N > 0$) and $g_1(Q_{ij})$ defined in Definition 55 are concave in $\{Q_{ij}\}$.

Proof: This can easily be proved, e.g., using the log-sum inequality [16]. □

The implications of the validity of Conjecture 74 for the generalized BAA were discussed at the end of Section IV-A, see also Footnote 6. Further, if the conjecture were true, it would probably also help in deriving termination conditions for the generalized BAA that are similar to the ones for the classical BAA in Lemma 11 and Remark 12.

VIII. Conclusion

In this paper, we considered the problem of generalizing the classical BAA to the problem of finding mutual-information-rate-maximizing parameters of an FSMS at the input to an indecomposable FSMC. To this end, we first formulated the classical BAA in such a way that it could be generalized to the new problem at hand. Then, we introduced the new algorithm, the generalized BAA, and characterized it by stating some local convergence properties.

Besides evaluating single-letter expressions like in the classical BAA, it seems unavoidable that the generalized BAA needs to evaluate more complicated expressions (T_{ij} values, see Definition 41). But it turns out that these expressions can be estimated very efficiently by evaluating some functions on a randomly generated input–output sequence pair. Indeed, the T_{ij} values are interesting quantities as they can capture infinitely long stretching memory, yet they can be estimated efficiently with an accuracy that is sufficient for any practical application (see also the corresponding comments about related accuracy issues mentioned in [19]).

While our results lead to a complete understanding of the local behavior of the generalized BAA, the global behavior is still not characterized. Studies on some channels suggest that the conjectures in Section VII are true, but we do not have a proof for the general case yet.

A side result of our investigations is a condition that characterizes information-rate-maximizing FSMSs for a given FSMC; we gave an intuitive explanation for it and showed how it generalizes the well-known condition for information-rate-maximizing DMSs for a given DMC.

Appendix A
Proof of Lemma 44

We have to maximize

$$\Psi\left(\tilde{Q}_{ij}, Q_{ij}, W\right) = \sum_{(i,j)\in\mathcal{B}} Q_{ij}\left(\log\left(\frac{\sum_{j'\in\vec{\mathcal{B}}_i} Q_{ij'}}{Q_{ij}}\right) + \tilde{T}_{ij}\right)$$

over $\{Q_{ij}\}$ under the constraints[11]

$$\sum_{(i,j)\in\mathcal{B}} Q_{ij} - 1 = 0$$

$$\sum_{k\in\overleftarrow{\mathcal{B}}_i} Q_{ki} - \sum_{j\in\vec{\mathcal{B}}_i} Q_{ij} = 0 \quad (\text{for all } i\in\mathcal{S}).$$

[11]For the moment, we neglect the constraints $Q_{ij} \geq 0$ for all $(i,j) \in \mathcal{B}$.

This is equivalent to setting the gradient of the Lagrangian[12]

$$\begin{aligned}L=&\sum_{(i,j)\in\mathcal{B}} Q_{ij}\left(\log\left(\frac{\sum_{j'\in\vec{\mathcal{B}}_i} Q_{ij'}}{Q_{ij}}\right)+\tilde{T}_{ij}\right)\\&+\lambda\left(\sum_{(i,j)\in\mathcal{B}} Q_{ij}-1\right)+\sum_{i\in\mathcal{S}}\lambda_i\left(\sum_{k\in\overleftarrow{\mathcal{B}}_i} Q_{ki}-\sum_{j\in\vec{\mathcal{B}}_i} Q_{ij}\right)\\=&\sum_{(i,j)\in\mathcal{B}} Q_{ij}\left(\log\left(\frac{\sum_{j'\in\vec{\mathcal{B}}_i} Q_{ij'}}{Q_{ij}}\right)+\tilde{T}_{ij}\right)\\&+\lambda\left(\sum_{(i,j)\in\mathcal{B}} Q_{ij}-1\right)\\&+\left(\sum_{(i,j)\in\mathcal{B}}\lambda_j Q_{ij}\right)-\left(\sum_{(i,j)\in\mathcal{B}}\lambda_i Q_{ij}\right)\end{aligned}$$

equal to zero

$$\begin{cases}\frac{\partial L}{\partial Q_{ij}}\stackrel{!}{=}0 & (\text{for all } (i,j)\in\mathcal{B})\\ \frac{\partial L}{\partial\lambda}\stackrel{!}{=}0 & \\ \frac{\partial L}{\partial\lambda_i}\stackrel{!}{=}0, & (\text{for all } i\in\mathcal{S}).\end{cases}$$

Solving these equations we get

$$\begin{aligned}0\stackrel{!}{=}\frac{\partial L}{\partial Q_{ij}}&=\left(\log\left(\frac{\sum_{j'\in\vec{\mathcal{B}}_i} Q_{ij'}}{Q_{ij}}\right)+\tilde{T}_{ij}\right)\\&\quad+\underbrace{\sum_{j''\in\vec{\mathcal{B}}_i} Q_{ij''}\frac{1}{\sum_{j'\in\vec{\mathcal{B}}_i} Q_{ij'}}}_{=1}-Q_{ij}\frac{1}{Q_{ij}}+\lambda+\lambda_j-\lambda_i\\&=-\log p_{ij}+\tilde{T}_{ij}+\lambda+\lambda_j-\lambda_i\ \ (\text{for all }\ (i,j)\in\mathcal{B})\\0\stackrel{!}{=}\frac{\partial L}{\partial\lambda}&=\sum_{(i,j)\in\mathcal{B}} Q_{ij}-1\\0\stackrel{!}{=}\frac{\partial L}{\partial\lambda_i}&=\sum_{k\in\overleftarrow{\mathcal{B}}_i} Q_{ki}-\sum_{j\in\vec{\mathcal{B}}_i} Q_{ij}\quad(\text{for all }\ i\in\mathcal{S}).\end{aligned}\tag{45}$$

This implies that

$$p_{ij}=\mathrm{e}^{\lambda_j-\lambda_i+\lambda+\tilde{T}_{ij}}\quad(\text{for all }\ (i,j)\in\mathcal{B}).\tag{46}$$

By the definition of the p_{ij}'s we must have $\sum_{j\in\vec{\mathcal{B}}_i} p_{ij}=1$ for all $i\in\mathcal{S}$. Therefore, we get

$$\sum_{j\in\vec{\mathcal{B}}_i}\mathrm{e}^{\tilde{T}_{ij}}\mathrm{e}^{\lambda_j}=\mathrm{e}^{-\lambda}\mathrm{e}^{\lambda_i}\quad(\text{for all }\ i\in\mathcal{S}).\tag{47}$$

Let $\tilde{\mathbf{A}}$ be the matrix with entries (see also Definition 43)

$$\tilde{\mathsf{A}}_{ij}\triangleq\begin{cases}\mathrm{e}^{\tilde{T}_{ij}} & (\text{if } (i,j)\in\mathcal{B})\\ 0 & (\text{otherwise})\end{cases}$$

and let $\tilde{\boldsymbol{\gamma}}$ be the vector with entries $\tilde{\gamma}_i\triangleq\mathrm{e}^{\lambda_i}$, and $\tilde{\rho}\triangleq\mathrm{e}^{-\lambda}$, then

$$\tilde{\mathbf{A}}\tilde{\boldsymbol{\gamma}}=\tilde{\rho}\cdot\tilde{\boldsymbol{\gamma}}\tag{48}$$

[12]Let us remark that Justesen and Høholdt [46] and Khayrallah and Neuhoff [24] have considered related optimization problems. However, instead of the summand $\sum_{(i,j)\in\mathcal{B}} Q_{ij}\tilde{T}_{ij}$ in the objective function, they have an additional upper bound of the type $\sum_{(i,j)\in\mathcal{B}} Q_{ij}w_{ij}\le W$. Because of this close connection, the Lagrangians associated to our and their maximization problems, respectively, are of course very much related.

i.e., $\tilde{\boldsymbol{\gamma}}$ must be a *right* eigenvector of $\tilde{\mathbf{A}}$ with a positive (and therefore also real) eigenvalue $\tilde{\rho}$. Moreover, all entries of $\tilde{\boldsymbol{\gamma}}$ must be positive (see also the comments in the paragraphs after (51)). Inserting these results into (46) we get

$$p_{ij}=\frac{\tilde{\gamma}_j}{\tilde{\gamma}_i}\cdot\frac{\tilde{\mathsf{A}}_{ij}}{\tilde{\rho}}\quad(\text{for all }\ (i,j)\in\mathcal{B}).\tag{49}$$

From $\sum_{i\in\overleftarrow{\mathcal{B}}_j}\mu_i p_{ij}=\mu_j$ (for all $j\in\mathcal{S}$) follows

$$\sum_{i\in\overleftarrow{\mathcal{B}}_j}\mu_i\frac{\tilde{\gamma}_j\tilde{\mathsf{A}}_{ij}}{\tilde{\gamma}_i\tilde{\rho}}=\mu_j\quad(\text{for all }\ j\in\mathcal{S})\tag{50}$$

and by letting the vector $\tilde{\boldsymbol{\beta}}$ have entries $\tilde{\beta}_i\triangleq\mu_i/(\tilde{K}\tilde{\gamma}_i)$ (where $\tilde{K}$ will be determined later) we obtain

$$\sum_{i\in\overleftarrow{\mathcal{B}}_j}\tilde{\beta}_i\tilde{\mathsf{A}}_{ij}=\tilde{\rho}\cdot\tilde{\beta}_j\quad(\text{for all }\ j\in\mathcal{S})$$

or, equivalently

$$\tilde{\boldsymbol{\beta}}^{\mathsf{T}}\tilde{\mathbf{A}}=\tilde{\rho}\cdot\tilde{\boldsymbol{\beta}}^{\mathsf{T}}$$

i.e., $\tilde{\boldsymbol{\beta}}^{\mathsf{T}}$ is a *left* eigenvector of $\tilde{\mathbf{A}}$ with eigenvalue $\tilde{\rho}$, and the entries of $\tilde{\boldsymbol{\beta}}$ must be nonnegative. Consequently, to fulfill $\sum_{i\in\mathcal{S}}\mu_i=1$, we must have $\mu_i=\tilde{K}\cdot\tilde{\beta}_i\cdot\tilde{\gamma}_i$ for all $i\in\mathcal{S}$, which implies $\tilde{K}\triangleq 1/\sum_{i\in\mathcal{S}}\tilde{\beta}_i\tilde{\gamma}_i$. Finally, we set $\mu_i^*\triangleq\mu_i$, $p_{ij}^*\triangleq p_{ij}$, and $Q_{ij}^*\triangleq\mu_i^* p_{ij}^*$.

We still have to determine what eigenvalue of $\tilde{\mathbf{A}}$ we have to take. But before investigating this issue, we would now like to show that $\Psi\left(\tilde{Q}_{ij},Q_{ij}^*,W\right)=\log(\tilde{\rho})$. This indeed follows from

$$\begin{aligned}&\Psi\left(\tilde{Q}_{ij},Q_{ij}^*,W\right)\\&=\sum_{(i,j)\in\mathcal{B}}\mu_i^* p_{ij}^*\left(\log\left(\frac{1}{p_{ij}^*}\right)+\tilde{T}_{ij}\right)\\&\stackrel{(a)}{=}\sum_{(i,j)\in\mathcal{B}}\mu_i^* p_{ij}^*\left(\log\left(\frac{1}{p_{ij}^*}\right)+\log(\tilde{\mathsf{A}}_{ij})\right)\\&=\sum_{(i,j)\in\mathcal{B}}\tilde{K}\tilde{\beta}_i\tilde{\gamma}_i\frac{\tilde{\gamma}_j\tilde{\mathsf{A}}_{ij}}{\tilde{\gamma}_i\tilde{\rho}}\left(\log\left(\frac{\tilde{\gamma}_i\tilde{\rho}}{\tilde{\gamma}_j\tilde{\mathsf{A}}_{ij}}\right)+\log(\tilde{\mathsf{A}}_{ij})\right)\\&=\frac{1}{\tilde{\rho}}\sum_{(i,j)\in\mathcal{B}}\tilde{K}\tilde{\beta}_i\tilde{\mathsf{A}}_{ij}\tilde{\gamma}_j\big(\log(\tilde{\gamma}_i)+\log(\tilde{\rho})-\log(\tilde{\gamma}_j)\big)\\&=\frac{1}{\tilde{\rho}}\left(\sum_{i\in\mathcal{S}}\tilde{K}\tilde{\beta}_i\big(\log(\tilde{\gamma}_i)+\log(\tilde{\rho})\big)\underbrace{\sum_{j\in\vec{\mathcal{B}}_i}\tilde{\mathsf{A}}_{ij}\tilde{\gamma}_j}_{=\tilde{\rho}\tilde{\gamma}_i}\right)\\&\quad-\frac{1}{\tilde{\rho}}\left(\sum_{j\in\mathcal{S}}\tilde{K}\tilde{\gamma}_j\log(\tilde{\gamma}_j)\underbrace{\sum_{i\in\overleftarrow{\mathcal{B}}_j}\tilde{\beta}_i\tilde{\mathsf{A}}_{ij}}_{=\tilde{\rho}\tilde{\beta}_j}\right)\\&=\sum_{i\in\mathcal{S}}\mu_i^*\big(\log(\tilde{\gamma}_i)+\log(\tilde{\rho})\big)-\sum_{j\in\mathcal{S}}\mu_j^*\log(\tilde{\gamma}_j)\\&=\log(\tilde{\rho})\end{aligned}\tag{51}$$

where at step (a) we used $\tilde{T}_{ij} = \log(\tilde{\mathsf{A}}_{ij})$ for $(i,j) \in \mathcal{B}$. The value $\log(\tilde{\rho})$ would clearly be maximized by taking $\tilde{\rho}$ to be the largest real eigenvalue of $\tilde{\mathbf{A}}$. But, as we have seen before, the right eigenvector corresponding to the eigenvalue $\tilde{\rho}$ must have positive entries and the left eigenvector must have nonnegative entries. The question is whether this can be fulfilled at all.

For an irreducible and nonnegative matrix $\tilde{\mathbf{A}}$ one can indeed show that these conditions can be met [54, p. 508].[13] One can show that such matrices have a real eigenvalue whose modulus is the largest of all eigenvalues. Moreover, it is an algebraically and geometrically single eigenvalue. (There may be other complex eigenvalues having the same modulus, though.) Such an eigenvalue, which is called the *Perron eigenvalue*, has a left and a right eigenvector whose entries are all positive. When their entries sum to one, respectively, one calls these eigenvectors the *left* and the *right* Perron eigenvector, respectively.

We come now shortly back to the comment in Footnote 11. From the above comments, since the Perron eigenvectors have positive entries, we must automatically have $Q^*_{ij} \geq 0$ for all $(i,j) \in \mathcal{B}$, i.e., neglecting these constraints in the first place was legal.

We now confirm that $\log(\tilde{\rho})$ is indeed the largest possible value for $\Psi(\tilde{Q}_{ij}, Q_{ij}, W)$ for given $\tilde{Q}_{ij}$ and W and varying Q_{ij}. Let $\{p^*_{ij}\}$ be the solution given in (49). For any $\{Q_{ij}\}$ we have

$$\begin{aligned}
&\Psi\left(\tilde{Q}_{ij}, Q^*_{ij}, W\right) - \Psi\left(\tilde{Q}_{ij}, Q_{ij}, W\right) \qquad (52)\\
&= \log(\tilde{\rho}) - \sum_{(i,j)\in\mathcal{B}} \mu_i p_{ij} \left(\log\left(\frac{1}{p_{ij}}\right) + \tilde{T}_{ij}\right)\\
&= \sum_{(i,j)\in\mathcal{B}} \mu_i p_{ij} \log\left(\frac{p_{ij}}{e^{\tilde{T}_{ij}}/\tilde{\rho}}\right)\\
&\overset{(a)}{=} \sum_{i\in\mathcal{S}} \mu_i \underbrace{\sum_{j\in\vec{\mathcal{B}}_i} p_{ij} \log\left(\frac{p_{ij}}{p^*_{ij}}\right)}_{\overset{(b)}{\geq} 0} + \sum_{(i,j)\in\mathcal{B}} \mu_i p_{ij} \log(\tilde{\gamma}_j)\\
&\quad - \sum_{(i,j)\in\mathcal{B}} \mu_i p_{ij} \log(\tilde{\gamma}_i)\\
&\geq \sum_{j\in\mathcal{S}} \log(\tilde{\gamma}_j) \underbrace{\sum_{i\in\overleftarrow{\mathcal{B}}_j} \mu_i p_{ij}}_{=\mu_j} - \sum_{i\in\mathcal{S}} \mu_i \log(\tilde{\gamma}_i) \underbrace{\sum_{j\in\vec{\mathcal{B}}_i} p_{ij}}_{=1}\\
&= \sum_{j\in\mathcal{S}} \mu_j \log(\tilde{\gamma}_j) - \sum_{i\in\mathcal{S}} \mu_i \log(\tilde{\gamma}_i) = 0 \qquad (53)
\end{aligned}$$

where at step (a) we used the fact that for $(i,j) \in \mathcal{A}$ we have $e^{\tilde{T}_{ij}}/\tilde{\rho} = p^*_{ij} \cdot \tilde{\gamma}_i/\tilde{\gamma}_j$ and (b) follows from the fact that relative entropies are nonnegative [16]. This proves our claim. We note that once given the correct solution, (52)–(53) are sufficient to show that this is also the optimal solution.

[13]The conditions in the lemma statement guarantee that $\tilde{\mathbf{A}}$ enjoys these properties.

Appendix B
Proof of Lemmas 53, 54, 57–60, 64, 70 and Theorems 65, 66

Proof of Lemma 53: Because of Assumption 50, we have $V(\boldsymbol{b}''|\boldsymbol{y}) > 0$ and because $V(\boldsymbol{b}, \boldsymbol{b}''|\boldsymbol{y}) = V(\boldsymbol{b}''|\boldsymbol{y})$, it follows that $V(\boldsymbol{b}, \boldsymbol{b}''|\boldsymbol{y}) > 0$ and that $V(\boldsymbol{b}, b''_\ell|\boldsymbol{y}) > 0$. Now (33) follows from

$$\begin{aligned}
V(\boldsymbol{b}|\boldsymbol{y}) &= \frac{V(\boldsymbol{b}|\boldsymbol{y})V(\boldsymbol{b}, \boldsymbol{b}''|\boldsymbol{y})}{V(\boldsymbol{b}, \boldsymbol{b}''|\boldsymbol{y})}\\
&= \frac{V(\boldsymbol{b}|\boldsymbol{y})V(\boldsymbol{b}''|\boldsymbol{y})}{V(\boldsymbol{b}, \boldsymbol{b}''|\boldsymbol{y})} \overset{(a)}{=} \frac{V(\boldsymbol{b}''|\boldsymbol{y})}{V(\boldsymbol{b}''|\boldsymbol{b}, \boldsymbol{y})}
\end{aligned}$$

where at step (a) we used $V(\boldsymbol{b}''|\boldsymbol{b}, \boldsymbol{y}) = V(\boldsymbol{b}, \boldsymbol{b}''|\boldsymbol{y})/V(\boldsymbol{b}|\boldsymbol{y})$. Equations (34) and (35) follow from

$$\begin{aligned}
&V(\boldsymbol{b}|\boldsymbol{y})\\
&\quad = \frac{V(\boldsymbol{b}|\boldsymbol{y})V(\boldsymbol{b}, b''_\ell|\boldsymbol{y})}{V(\boldsymbol{b}, b''_\ell|\boldsymbol{y})} \overset{(a)}{=} \frac{V(\boldsymbol{b}, b''_\ell|\boldsymbol{y})}{V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})}\\
&\quad \overset{(b)}{=} \frac{V(b''_\ell|\boldsymbol{y}) \cdot V(\boldsymbol{b}^N_{\ell+1}|s''_\ell, \boldsymbol{y}^N_{\ell+1}) \cdot V(\boldsymbol{b}^{\ell-1}_{-N+1}|s''_{\ell-1}, \boldsymbol{y}^{\ell-1}_{-N+1})}{V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})}\\
&V(\boldsymbol{b}|\boldsymbol{y})\\
&\quad = \frac{V(\boldsymbol{b}|\boldsymbol{y})V(\boldsymbol{b}, s''_\ell|\boldsymbol{y})}{V(\boldsymbol{b}, s''_\ell|\boldsymbol{y})} \overset{(a)}{=} \frac{V(\boldsymbol{b}, s''_\ell|\boldsymbol{y})}{V(s''_\ell|\boldsymbol{b}, \boldsymbol{y})}\\
&\quad \overset{(b)}{=} \frac{V(s''_\ell|\boldsymbol{y}) \cdot V(\boldsymbol{b}^N_{\ell+1}|s''_\ell, \boldsymbol{y}^N_{\ell+1}) \cdot V(\boldsymbol{b}^{\ell}_{-N+1}|s''_{\ell}, \boldsymbol{y}^{\ell}_{-N+1})}{V(s''_\ell|\boldsymbol{b}, \boldsymbol{y})}
\end{aligned}$$

where at steps (a) we used $V(b''_\ell|\boldsymbol{b}, \boldsymbol{y}) = V(\boldsymbol{b}, b''_\ell|\boldsymbol{y})/V(\boldsymbol{b}|\boldsymbol{y})$ and $V(s''_\ell|\boldsymbol{b}, \boldsymbol{y}) = V(\boldsymbol{b}, s''_\ell|\boldsymbol{y})/V(\boldsymbol{b}|\boldsymbol{y})$, respectively, and at steps (b) we used the Markov property for *a posteriori* pmfs shown in Lemma 52. Equation (36) follows from

$$\begin{aligned}
V(\boldsymbol{b}''|\boldsymbol{y}) &\overset{(a)}{=} V(s''_{-N}|\boldsymbol{y}) \cdot \prod_{\ell\in\mathcal{I}_N} V(b''_\ell|s''_{\ell-1}, \boldsymbol{y})\\
&= V(s''_{-N}|\boldsymbol{y}) \cdot \prod_{\ell\in\mathcal{I}_N} \frac{V(b''_\ell|\boldsymbol{y})}{V(s''_{\ell-1}|\boldsymbol{y})}\\
&= \left(\prod_{\ell\in\mathcal{I}_N} V(b''_\ell|\boldsymbol{y})\right) \cdot \left(\prod_{\ell\in\mathcal{I}'_N} V(s''_\ell|\boldsymbol{y})\right)^{-1}
\end{aligned}$$

where equality (a) follows from Lemma 52. Similar steps lead to (37).

Proof of Lemma 54:

Remark: Although the following derivation might look quite lengthy, the idea behind it is rather simple. To show the idea, we look at a simplified example. So, let

$$h(\alpha) \triangleq h_1(\alpha) \cdot h_2(\alpha)/h_3(\alpha)$$

be a function of α. Then

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\alpha} h(\alpha) &= \frac{h_2(\alpha)}{h_3(\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}\alpha} h_1(\alpha)\right) + \frac{h_1(\alpha)}{h_3(\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}\alpha} h_2(\alpha)\right)\\
&\quad - \frac{h_1(\alpha)h_2(\alpha)}{h_3^2(\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}\alpha} h_3(\alpha)\right)
\end{aligned}$$

$$= \frac{h(\alpha)}{h_1(\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}\alpha} h_1(\alpha) \right) + \frac{h(\alpha)}{h_2(\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}\alpha} h_2(\alpha) \right) - \frac{h(\alpha)}{h_3(\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}\alpha} h_3(\alpha) \right).$$

Fix some legal branch sequence $\boldsymbol{b}$ and let $\boldsymbol{s} \triangleq \boldsymbol{s}(\boldsymbol{b})$. In order to prove the lemma, we start with $Q(\boldsymbol{b})$ as given in (11), i.e.,

$$Q(\boldsymbol{b}) = \frac{\prod_{\ell \in \mathcal{I}_N} Q_{b_\ell}}{\prod_{\ell \in \mathcal{I}'_N} \mu_{s_\ell}}$$

or logarithmically

$$\begin{aligned} \log(Q(\boldsymbol{b})) &= \left(\sum_{\ell \in \mathcal{I}_N} \log(Q_{b_\ell}) \right) - \left(\sum_{\ell \in \mathcal{I}'_N} \log(\mu_{s_\ell}) \right) \\ &= \left(\sum_{(i,j) \in \mathcal{B}} \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \log(Q_{b_\ell}) \right) - \left(\sum_{i \in \mathcal{S}} \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \log(\mu_{s_\ell}) \right) \\ &= \left(\sum_{(i,j) \in \mathcal{B}} \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \log(Q_{ij}) \right) - \left(\sum_{i \in \mathcal{S}} \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \log(\mu_i) \right). \end{aligned}$$

Taking the derivative with respect to α, we obtain

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\alpha} \log(Q(\boldsymbol{b})) \\ &\quad = \left(\sum_{(i,j) \in \mathcal{B}} \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \frac{1}{Q_{ij}} Q_{ij}^\alpha \right) - \left(\sum_{i \in \mathcal{S}} \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \frac{1}{\mu_i} \mu_i^\alpha \right) \\ &\quad = \left(\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \frac{1}{Q_{ij}} \right) - \left(\sum_{i \in \mathcal{S}} \mu_i^\alpha \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \frac{1}{\mu_i} \right). \end{aligned}$$

But from $\frac{\mathrm{d}}{\mathrm{d}\alpha} \log\big(Q(\boldsymbol{b})\big) = \big(\frac{\mathrm{d}}{\mathrm{d}\alpha} Q(\boldsymbol{b})\big) / Q(\boldsymbol{b})$ it follows that $\frac{\mathrm{d}}{\mathrm{d}\alpha} Q(\boldsymbol{b}) = Q(\boldsymbol{b}) \cdot \frac{\mathrm{d}}{\mathrm{d}\alpha} \log(Q(\boldsymbol{b}))$ and so, using (26) in step (a)

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\alpha} Q(\boldsymbol{b}) \\ &\quad = \left(\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \frac{Q(\boldsymbol{b})}{Q_{ij}} \right) - \left(\sum_{i \in \mathcal{S}} \mu_i^\alpha \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \frac{Q(\boldsymbol{b})}{\mu_i} \right) \\ &\quad \overset{\text{(a)}}{=} \left(\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \sum_{\substack{\ell \in \mathcal{I}_N \\ b_\ell = (i,j)}} \frac{Q(\boldsymbol{b})}{Q_{ij}} \right) - \left(\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \sum_{\substack{\ell \in \mathcal{I}'_N \\ s_\ell = i}} \frac{Q(\boldsymbol{b})}{\mu_i} \right). \end{aligned}$$

Proof of Lemma 57: We prove the statement in the lemma by using

$$H(B_{-N+1}, \ldots, B_N) = H(S_{-N}) + \sum_{\ell \in \mathcal{I}_N} H(B_\ell | S_{\ell-1}).$$

Together with the time invariance of $H(S_{-N})$ and $H(B_\ell | S_{\ell-1})$ (which is guaranteed by our choice of the joint pmf in Definition 35), this implies

$$\begin{aligned} &-\frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \log Q(\boldsymbol{b}) \\ &\quad = -\frac{1}{2N} \sum_{i \in \mathcal{S}} \mu_i \log(\mu_i) - \sum_{(i,j) \in \mathcal{B}} Q_{ij} \log(p_{ij}) \\ &\quad = -\frac{1}{2N} \sum_{(i,j) \in \mathcal{B}} Q_{ij} \log(\mu_i) - \sum_{(i,j) \in \mathcal{B}} Q_{ij} \log(p_{ij}). \end{aligned}$$

Proof of Lemma 58: We take the expression for $g_1^{(N)}(Q_{ij})$ from Lemma 57 and differentiate it with respect to α

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha} g_1^{(N)}(\alpha) &= -\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \log(p_{ij}) - \sum_{(i,j) \in \mathcal{B}} Q_{ij} \frac{1}{p_{ij}} p_{ij}^\alpha \\ &\quad - \frac{1}{2N} \sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \log(\mu_i) - \frac{1}{2N} \sum_{(i,j) \in \mathcal{B}} Q_{ij} \frac{1}{\mu_i} \mu_i^\alpha \\ &= -\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \log(p_{ij}) - \sum_{i \in \mathcal{S}} \mu_i \sum_{j \in \vec{\mathcal{B}}_i} p_{ij}^\alpha \\ &\quad - \frac{1}{2N} \sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \log(\mu_i) - \frac{1}{2N} \sum_{i \in \mathcal{S}} \mu_i^\alpha \underbrace{\sum_{j \in \vec{\mathcal{B}}_i} p_{ij}}_{=1} \\ &\overset{\text{(a)}}{=} -\sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \log(p_{ij}) - \frac{1}{2N} \sum_{(i,j) \in \mathcal{B}} Q_{ij}^\alpha \log(\mu_i) \end{aligned}$$

where step (a) follows from (24) and (25).

Proof of Lemma 59: We rewrite the function $g_2^{(N)}(Q_{ij}, W)$ given in Definition 55 using the T_{ij} values from Definition 41 as shown in the equation at the bottom of the following page, where at step (a) we have used (33), at step (b) we have used (36) and (37), and where at step (c) we have used (8)–(9).

Proof of Lemma 60:

Note: For any $\boldsymbol{b}$ and any $\boldsymbol{y}$ the value of $W(\boldsymbol{y}|\boldsymbol{b})$ is independent of α. But one must be very careful when differentiating $V(\boldsymbol{b}|\boldsymbol{y})$ with respect to α, as $V(\boldsymbol{b}|\boldsymbol{y})$ depends on $Q(\boldsymbol{b})$, which depends on Q_{ij}, which depends on α, see also (11)–(12).

Differentiating

$$\begin{aligned} -g_2^{(N)}(\alpha, W) &= \frac{1}{2N} \sum_{\boldsymbol{b}} \sum_{\boldsymbol{y}} Q(\boldsymbol{b}) W(\boldsymbol{y}|\boldsymbol{b}) \log(V(\boldsymbol{b}|\boldsymbol{y})) \\ &= \frac{1}{2N} \sum_{\boldsymbol{b}} \sum_{\boldsymbol{y}} Q(\boldsymbol{b}) W(\boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{Q(\boldsymbol{b}) W(\boldsymbol{y}|\boldsymbol{b})}{R(\boldsymbol{y})} \right) \end{aligned}$$

with respect to α yields (in the following, we will use the abbreviation $J \triangleq -\frac{\mathrm{d}}{\mathrm{d}\alpha} g_2^{(N)}(\alpha, W)$) the equation at the top of the following page, where equality (a) follows from (38). Using (34) and (35), we continue to evaluate J, namely, see the second equation at the top of the following page.

However, this is equal to the equation at the top of the subsequent page. In the following, we use $\overline{\overline{T}}_{ij}^{(N)}$, $\overline{T}_i^{(N)}$, $\overline{\overline{T}}_{ij}^{(N)}(\ell)$, and $\overline{T}_i^{(N)}(\ell)$ as given in Definition 41. Additionally, for all $i \in \mathcal{S}$ and all $\ell \in \{-N, \ldots, +N\}$ we define $\overrightarrow{\chi}_i(\ell)$, $\overleftarrow{\chi}_i(\ell)$, and $\overrightarrow{\chi}_i$ in (54)-(56), also at the top of the subsequent page. These expressions help us to analyze the above sum representing J.

- The first term is equal to
$$\frac{1}{2N} \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \sum_{\ell\in\mathcal{I}_N} \overline{\overline{T}}_{ij}^{(N)}(\ell) = \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \overline{\overline{T}}_{ij}^{(N)};$$
- the second term is equal to $\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \sum_{\ell\in\mathcal{I}_N} \overrightarrow{\chi}_j(\ell)$;
- the third term is equal to $\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \sum_{\ell\in\mathcal{I}_N} \overleftarrow{\chi}_i(\ell-1)$;
- the fourth term is equal to
$$-\frac{1}{2N} \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \sum_{\ell\in\mathcal{I}'_N} \overline{T}_i(\ell) = -\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \overline{T}_i^{(N)};$$
- the fifth term is equal to $-\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \sum_{\ell\in\mathcal{I}'_N} \overrightarrow{\chi}_i(\ell)$;
- the sixth term is equal to $-\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \sum_{\ell\in\mathcal{I}'_N} \overleftarrow{\chi}_i(\ell)$.

Therefore, we can rewrite J as shown in (57) at the bottom of the subsequent page, where equality (a) follows from $\overrightarrow{\chi}_i(N) = 0$ and $\overleftarrow{\chi}_i(-N) = 0$ for all $i \in \mathcal{S}$, and equality (b) follows from (56). The auxiliary result

$$\sum_{(i,j)\in\mathcal{B}} \left(\overrightarrow{\chi}_j - \overrightarrow{\chi}_i\right) \cdot Q_{ij}^{\alpha} = \left(\sum_{j\in\mathcal{S}} \overrightarrow{\chi}_j \sum_{i\in\overleftarrow{\mathcal{B}}_j} Q_{ij}^{\alpha}\right) - \left(\sum_{i\in\mathcal{S}} \overrightarrow{\chi}_i \cdot \sum_{j\in\overrightarrow{\mathcal{B}}_i} Q_{ij}^{\alpha}\right)$$
$$\stackrel{(a)}{=} \left(\sum_{j\in\mathcal{S}} \overrightarrow{\chi}_j \cdot \mu_j^{\alpha}\right) - \left(\sum_{i\in\mathcal{S}} \overrightarrow{\chi}_i \cdot \mu_i^{\alpha}\right) = 0$$

where equality (a) follows from (26), helps us to simplify (57) even more to

$$J = \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha} \cdot T_{ij}^{(N)}.$$

Remembering that we set $J \triangleq -\frac{\mathrm{d}}{\mathrm{d}\alpha} g_2^{(N)}(\alpha, W)$, we have proved the lemma.

Proof of Lemma 64: From Definition 62 we have for some fixed $\tilde{\alpha}$

$$g_2'^{(N)}(\tilde{\alpha}, \alpha, W) = -\sum_{(i,j)\in\mathcal{B}} Q_{ij}(\alpha) \cdot T_{ij}^{(N)}(\tilde{\alpha}).$$

$$\begin{aligned}
-g_2^{(N)}(Q_{ij}, W) &= \frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{y}} W(\boldsymbol{y}|\boldsymbol{b}) \log\left(V(\boldsymbol{b}|\boldsymbol{y})\right) \\
&= \frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(V(\boldsymbol{b}|\boldsymbol{y})\right) \\
&\stackrel{(a)}{=} \frac{1}{2N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{V(\boldsymbol{b}''|\boldsymbol{y})}{V(\boldsymbol{b}''|\boldsymbol{b}, \boldsymbol{y})}\right) \\
&\stackrel{(b)}{=} \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{V(b''_\ell|\boldsymbol{y})}{V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})}\right) \\
&\quad - \frac{1}{2N} \sum_{\ell\in\mathcal{I}'_N} \sum_{\boldsymbol{b}} Q(\boldsymbol{b}) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{V(s''_\ell|\boldsymbol{y})}{V(s''_\ell|\boldsymbol{b}, \boldsymbol{y})}\right) \\
&= \sum_{(i,j)\in\mathcal{B}} Q_{ij} \frac{1}{2N} \sum_{\ell\in\mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell=(i,j)}} Q(\boldsymbol{b}|b_\ell) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{V(b''_\ell|\boldsymbol{y})}{V(b''_\ell|\boldsymbol{b}, \boldsymbol{y})}\right) \\
&\quad - \sum_{i\in\mathcal{S}} \mu_i \frac{1}{2N} \sum_{\ell\in\mathcal{I}'_N} \sum_{\substack{\boldsymbol{b} \\ s_\ell=i}} Q(\boldsymbol{b}|s_\ell) \sum_{\boldsymbol{b}'} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(\frac{V(s''_\ell|\boldsymbol{y})}{V(s''_\ell|\boldsymbol{b}, \boldsymbol{y})}\right) \\
&= \sum_{(i,j)\in\mathcal{B}} Q_{ij} \overline{\overline{T}}_{ij}^{(N)} - \sum_{i\in\mathcal{S}} \mu_i \overline{T}_i^{(N)} \\
&\stackrel{(c)}{=} \sum_{(i,j)\in\mathcal{B}} Q_{ij} \overline{\overline{T}}_{ij}^{(N)} - \sum_{(i,j)\in\mathcal{B}} Q_{ij} \overline{T}_i^{(N)} \\
&= \sum_{(i,j)\in\mathcal{B}} Q_{ij} \cdot T_{ij}^{(N)}
\end{aligned}$$

$$
\begin{aligned}
J &= \frac{1}{2N}\sum_{\boldsymbol{b}}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})\log\left(\frac{Q(\boldsymbol{b})W(\boldsymbol{y}|\boldsymbol{b})}{R(\boldsymbol{y})}\right)+\frac{1}{2N}\sum_{\boldsymbol{b}}Q(\boldsymbol{b})\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})\frac{1}{Q(\boldsymbol{b})}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\\
&\quad-\frac{1}{2N}\sum_{\boldsymbol{b}}Q(\boldsymbol{b})\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})\frac{1}{R(\boldsymbol{y})}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}R(\boldsymbol{y})\right)\\
&=\frac{1}{2N}\sum_{\boldsymbol{b}}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}|\boldsymbol{y})\right)+\frac{1}{2N}\sum_{\boldsymbol{b}}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\underbrace{\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})}_{=1}-\frac{1}{2N}\sum_{\boldsymbol{y}}\frac{R(\boldsymbol{y})}{R(\boldsymbol{y})}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}R(\boldsymbol{y})\right)\\
&=\frac{1}{2N}\sum_{\boldsymbol{b}}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}|\boldsymbol{y})\right)+\frac{1}{2N}\frac{\mathrm{d}}{\mathrm{d}\alpha}\underbrace{\sum_{\boldsymbol{b}}Q(\boldsymbol{b})}_{=1}-\frac{1}{2N}\frac{\mathrm{d}}{\mathrm{d}\alpha}\underbrace{\sum_{\boldsymbol{y}}R(\boldsymbol{y})}_{=1}\\
&=\frac{1}{2N}\sum_{\boldsymbol{b}}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\sum_{\boldsymbol{y}}W(\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}|\boldsymbol{y})\right)=\frac{1}{2N}\sum_{\boldsymbol{b}}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}Q(\boldsymbol{b})\right)\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}|\boldsymbol{y})\right)\\
&\overset{(a)}{=}\frac{1}{2N}\sum_{\boldsymbol{b}}\left[\left(\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}_N\\ b_\ell=(i,j)}}\frac{Q(\boldsymbol{b})}{Q_{ij}}\right)-\left(\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}'_N\\ s_\ell=i}}\frac{Q(\boldsymbol{b})}{\mu_i}\right)\right]\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}|\boldsymbol{y})\right)
\end{aligned}
$$

$$
\begin{aligned}
J &= \frac{1}{2N}\sum_{\boldsymbol{b}}\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}_N\\ b_\ell=(i,j)}}\frac{Q(\boldsymbol{b})}{Q_{ij}}\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(\frac{V(b'_\ell|\boldsymbol{y})}{V(b'_\ell|\boldsymbol{b},\boldsymbol{y})}\right)\\
&\quad+\frac{1}{2N}\sum_{\boldsymbol{b}}\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}_N\\ b_\ell=(i,j)}}\frac{Q(\boldsymbol{b})}{Q_{ij}}\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}_{\ell+1}^{N}|s'_\ell,\boldsymbol{y}_{\ell+1}^{N})\right)\\
&\quad+\frac{1}{2N}\sum_{\boldsymbol{b}}\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}_N\\ b_\ell=(i,j)}}\frac{Q(\boldsymbol{b})}{Q_{ij}}\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}_{-N+1}^{\ell-1}|s'_{\ell-1},\boldsymbol{y}_{-N+1}^{\ell-1})\right)\\
&\quad-\frac{1}{2N}\sum_{\boldsymbol{b}}\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}'_N\\ s_\ell=i}}\frac{Q(\boldsymbol{b})}{\mu_i}\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(\frac{V(s'_\ell|\boldsymbol{y})}{V(s'_\ell|\boldsymbol{b},\boldsymbol{y})}\right)\\
&\quad-\frac{1}{2N}\sum_{\boldsymbol{b}}\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}'_N\\ s_\ell=i}}\frac{Q(\boldsymbol{b})}{\mu_i}\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}_{\ell+1}^{N}|s'_\ell,\boldsymbol{y}_{\ell+1}^{N})\right)\\
&\quad-\frac{1}{2N}\sum_{\boldsymbol{b}}\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}\sum_{\substack{\ell\in\mathcal{I}'_N\\ s_\ell=i}}\frac{Q(\boldsymbol{b})}{\mu_i}\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}}W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(V(\boldsymbol{b}_{-N+1}^{\ell}|s'_\ell,\boldsymbol{y}_{-N+1}^{\ell})\right).
\end{aligned}
$$

Because the $T_{ij}^{(N)}(\tilde{\alpha})$'s are independent of α, we easily get

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\alpha}g_2'^{(N)}(\tilde{\alpha},\alpha,W)\Big|_{\alpha=\tilde{\alpha}} &= -\frac{\mathrm{d}}{\mathrm{d}\alpha}\sum_{(i,j)\in\mathcal{B}}Q_{ij}(\alpha)\cdot T_{ij}^{(N)}(\tilde{\alpha})\Big|_{\alpha=\tilde{\alpha}}\\
&= -\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}(\alpha)\cdot T_{ij}^{(N)}(\tilde{\alpha})\Big|_{\alpha=\tilde{\alpha}}\\
&= -\sum_{(i,j)\in\mathcal{B}}Q_{ij}^{\alpha}(\tilde{\alpha})\cdot T_{ij}^{(N)}(\tilde{\alpha}).
\end{aligned}
$$

This expression is equivalent to $\frac{\mathrm{d}}{\mathrm{d}\alpha}g_2^{(N)}(\alpha,W)$ (as given in Lemma 60) evaluated at $\alpha=\tilde{\alpha}$.

Proof of Theorem 65: From Remark 56 we have

$$I^{(N)}(Q_{ij},W)=g_1^{(N)}(Q_{ij})-g_2^{(N)}(Q_{ij},W).$$

Using Lemmas 57 and 59 this turns into

$$
\begin{aligned}
&I^{(N)}(Q_{ij},W)\\
&\quad=-\sum_{(i,j)\in\mathcal{B}}Q_{ij}\log\left(p_{ij}\right)-\frac{1}{2N}\sum_{i\in\mathcal{S}}\mu_i\log\left(\mu_i\right)\\
&\qquad+\sum_{(i,j)\in\mathcal{B}}Q_{ij}\cdot T_{ij}^{(N)}\\
&\quad\overset{(a)}{=}\sum_{(i,j)\in\mathcal{B}}Q_{ij}\cdot\left[-\log\left(p_{ij}\right)-\frac{1}{2N}\log\left(\mu_i\right)+T_{ij}^{(N)}\right]
\end{aligned}
$$

$$
\begin{aligned}
J = &\frac{1}{2N}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\sum_{\ell\in\mathcal{I}_N}\sum_{\substack{\boldsymbol{b}\\ b_\ell=(i,j)}} Q(\boldsymbol{b}|b_\ell)\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}} W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(\frac{V(b'_\ell|\boldsymbol{y})}{V(b'_\ell|\boldsymbol{b},\boldsymbol{y})}\right)\\
&+\frac{1}{2N}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\sum_{\ell\in\mathcal{I}_N}\sum_{\substack{\boldsymbol{b}_{\ell+1}^N\\ s_\ell=j}} Q(\boldsymbol{b}_{\ell+1}^N|s_\ell)\sum_{\boldsymbol{b}'^N_{\ell+1}}\sum_{\boldsymbol{y}_{\ell+1}^N} W(\boldsymbol{b}'^N_{\ell+1},\boldsymbol{y}_{\ell+1}^N|\boldsymbol{b}_{\ell+1}^N)\log\left(V(\boldsymbol{b}_{\ell+1}^N|s'_\ell,\boldsymbol{y}_{\ell+1}^N)\right)\\
&+\frac{1}{2N}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\sum_{\ell\in\mathcal{I}_N}\sum_{\substack{\boldsymbol{b}_{-N+1}^{\ell-1}\\ s_{\ell-1}=i}} Q(\boldsymbol{b}_{-N+1}^{\ell-1}|s_{\ell-1})\sum_{\boldsymbol{b}'^{\ell-1}_{-N+1}}\sum_{\boldsymbol{y}_{-N+1}^{\ell-1}} W(\boldsymbol{b}''^{\ell-1}_{-N+1},\boldsymbol{y}_{-N+1}^{\ell-1}|\boldsymbol{b}_{-N+1}^{\ell-1})\log\left(V(\boldsymbol{b}_{-N+1}^{\ell-1}|s''_{\ell-1},\boldsymbol{y}_{-N+1}^{\ell-1})\right)\\
&-\frac{1}{2N}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\sum_{\ell\in\mathcal{I}'_N}\sum_{\substack{\boldsymbol{b}\\ s_\ell=i}} Q(\boldsymbol{b}|s_\ell)\sum_{\boldsymbol{b}'}\sum_{\boldsymbol{y}} W(\boldsymbol{b}',\boldsymbol{y}|\boldsymbol{b})\log\left(\frac{V(s'_\ell|\boldsymbol{y})}{V(s'_\ell|\boldsymbol{b},\boldsymbol{y})}\right)\\
&-\frac{1}{2N}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\sum_{\ell\in\mathcal{I}'_N}\sum_{\substack{\boldsymbol{b}_{\ell+1}^N\\ s_\ell=i}} Q(\boldsymbol{b}_{\ell+1}^N|s_\ell)\sum_{\boldsymbol{b}'^N_{\ell+1}}\sum_{\boldsymbol{y}_{\ell+1}^N} W(\boldsymbol{b}'^N_{\ell+1},\boldsymbol{y}_{\ell+1}^N|\boldsymbol{b}_{\ell+1}^N)\log\left(V(\boldsymbol{b}_{\ell+1}^N|s'_\ell,\boldsymbol{y}_{\ell+1}^N)\right)\\
&-\frac{1}{2N}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\sum_{\ell\in\mathcal{I}'_N}\sum_{\substack{\boldsymbol{b}_{-N+1}^{\ell}\\ s_\ell=i}} Q(\boldsymbol{b}_{-N+1}^{\ell}|s_\ell)\sum_{\boldsymbol{b}'^{\ell}_{-N+1}}\sum_{\boldsymbol{y}_{-N+1}^{\ell}} W(\boldsymbol{b}''^{\ell}_{-N+1},\boldsymbol{y}_{-N+1}^{\ell}|\boldsymbol{b}_{-N+1}^{\ell})\log\left(V(\boldsymbol{b}_{-N+1}^{\ell}|s''_\ell,\boldsymbol{y}_{-N+1}^{\ell})\right).
\end{aligned}
$$

$$
\overrightarrow{\chi}_i(\ell) \triangleq \frac{1}{2N}\sum_{\substack{\boldsymbol{b}_{\ell+1}^N\\ s_\ell=i}} Q(\boldsymbol{b}_{\ell+1}^N|s_\ell)\sum_{\boldsymbol{b}'^N_{\ell+1}}\sum_{\boldsymbol{y}_{\ell+1}^N} W(\boldsymbol{b}'^N_{\ell+1},\boldsymbol{y}_{\ell+1}^N|\boldsymbol{b}_{\ell+1}^N)\log\left(V(\boldsymbol{b}_{\ell+1}^N|s'_\ell,\boldsymbol{y}_{\ell+1}^N)\right) \tag{54}
$$

$$
\overleftarrow{\chi}_i(\ell) \triangleq \frac{1}{2N}\sum_{\substack{\boldsymbol{b}_{-N+1}^{\ell}\\ s_\ell=i}} Q(\boldsymbol{b}_{-N+1}^{\ell}|s_\ell)\sum_{\boldsymbol{b}'^{\ell}_{-N+1}}\sum_{\boldsymbol{y}_{-N+1}^{\ell}} W(\boldsymbol{b}''^{\ell}_{-N+1},\boldsymbol{y}_{-N+1}^{\ell}|\boldsymbol{b}_{-N+1}^{\ell})\log\left(V(\boldsymbol{b}_{-N+1}^{\ell}|s''_\ell,\boldsymbol{y}_{-N+1}^{\ell})\right) \tag{55}
$$

$$
\overrightarrow{\chi}_i \triangleq \sum_{\ell\in\mathcal{I}'_N}\overrightarrow{\chi}_i(\ell). \tag{56}
$$

where at equality (a) we used that

$$
\sum_{i\in\mathcal{S}}\mu_i\log(\mu_i) = \sum_{(i,j)\in\mathcal{B}} Q_{ij}\log(\mu_i).
$$

Proof of Theorem 66: From Remark 56 we have

$$
I^{(N)}(Q_{ij},W) = g_1^{(N)}(Q_{ij}) - g_2^{(N)}(Q_{ij},W)
$$

therefore

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\alpha}I^{(N)}(Q_{ij},W)\\
&\quad=\frac{\mathrm{d}}{\mathrm{d}\alpha}g_1^{(N)}(Q_{ij}) - \frac{\mathrm{d}}{\mathrm{d}\alpha}g_2^{(N)}(Q_{ij},W)\\
&\quad\overset{(a)}{=}\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\cdot\left[-\log(p_{ij}) - \frac{1}{2N}\log(\mu_i)\right]\\
&\qquad+\sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}(\alpha)\cdot T_{ij}^{(N)}
\end{aligned}
$$

$$
\begin{aligned}
J &= \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\left[\overline{\overline{T}}_{ij}^{(N)} - \overline{T}_i^{(N)} + \sum_{\ell\in\mathcal{I}_N}\overrightarrow{\chi}_j(\ell) - \sum_{\ell\in\mathcal{I}'_N}\overrightarrow{\chi}_i(\ell) + \sum_{\ell\in\mathcal{I}_N}\overleftarrow{\chi}_i(\ell-1) - \sum_{\ell\in\mathcal{I}'_N}\overleftarrow{\chi}_i(\ell)\right]\\
&= \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\left[T_{ij}^{(N)} + \sum_{\ell=-N+1}^{N}\overrightarrow{\chi}_j(\ell) - \sum_{\ell=-N+1}^{N-1}\overrightarrow{\chi}_i(\ell) + \sum_{\ell=-N}^{N-1}\overleftarrow{\chi}_i(\ell) - \sum_{\ell=-N+1}^{N-1}\overleftarrow{\chi}_i(\ell)\right]\\
&\overset{(a)}{=} \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\left[T_{ij}^{(N)} + \sum_{\ell=-N+1}^{N-1}\overrightarrow{\chi}_j(\ell) - \sum_{\ell=-N+1}^{N-1}\overrightarrow{\chi}_i(\ell)\right]\\
&\overset{(b)}{=} \sum_{(i,j)\in\mathcal{B}} Q_{ij}^{\alpha}\left[T_{ij}^{(N)} + \overrightarrow{\chi}_j - \overrightarrow{\chi}_i\right]
\end{aligned} \tag{57}
$$

$$\overline{\overline{T}}_{ij}^{(N)} = \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell = (i,j)}} Q(\boldsymbol{b}|b_\ell) \sum_{\boldsymbol{b}''} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(V(b_\ell''|\boldsymbol{y})\right) - \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell = (i,j)}} Q(\boldsymbol{b}|b_\ell) \sum_{\boldsymbol{b}''} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(V(b_\ell''|\boldsymbol{b}, \boldsymbol{y})\right).$$

$$\begin{aligned}
\eta_1 &= \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell = (i,j)}} Q(\boldsymbol{b}|b_\ell) \sum_{\boldsymbol{b}''} \sum_{\boldsymbol{y}} W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b}) \log\left(V(b_\ell''|\boldsymbol{y})\right) \\
&= \sum_{\boldsymbol{y}} R(\boldsymbol{y}) \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell = (i,j)}} \sum_{\boldsymbol{b}''} \frac{Q(\boldsymbol{b}|b_\ell) W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b})}{R(\boldsymbol{y})} \log\left(V(b_\ell''|\boldsymbol{y})\right) \\
&= \sum_{\boldsymbol{y}} R(\boldsymbol{y}) \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{\boldsymbol{b} \\ b_\ell = (i,j)}} \sum_{\boldsymbol{b}''} \frac{Q(\boldsymbol{b}) W(\boldsymbol{b}'', \boldsymbol{y}|\boldsymbol{b})}{Q_{ij} R(\boldsymbol{y})} \log\left(V(b_\ell''|\boldsymbol{y})\right) \\
&= \sum_{\boldsymbol{y}} R(\boldsymbol{y}) \left[\frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{b_\ell'' \\ b_\ell = (i,j)}} \frac{V(b_\ell''|\boldsymbol{y})}{Q_{ij}} \log\left(V(b_\ell''|\boldsymbol{y})\right) \right].
\end{aligned}$$

$$\eta_2 = -\sum_{\boldsymbol{b}} \sum_{\boldsymbol{y}} Q(\boldsymbol{b}) W(\boldsymbol{y}|\boldsymbol{b}) \left[\frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{b_\ell'' \\ b_\ell = (i,j)}} \frac{V(b_\ell''|\boldsymbol{b}, \boldsymbol{y})}{Q_{ij}} \log\left(V(b_\ell''|\boldsymbol{b}, \boldsymbol{y})\right) \right].$$

$$\overline{\check{\overline{T}}}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}}) \approx \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{b_\ell'' \\ b_\ell = (i,j)}} \frac{V(b_\ell''|\check{\boldsymbol{y}})}{Q_{ij}} \log\left(V(b_\ell''|\check{\boldsymbol{y}})\right) - \frac{1}{2N} \sum_{\ell \in \mathcal{I}_N} \sum_{\substack{b_\ell'' \\ b_\ell = \check{b}_\ell = (i,j)}} \frac{V(b_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})}{Q_{ij}} \log\left(V(b_\ell''|\check{\boldsymbol{b}}, \check{\boldsymbol{y}})\right)$$

$$= \sum_{(i,j) \in \mathcal{B}} Q_{ij}^{\alpha} \cdot \left[-\log(p_{ij}) - \frac{1}{2N} \log(\mu_i) + T_{ij}^{(N)} \right]$$

where equality (a) follows from Lemmas 58 and 60.

Proof of Lemma 70: In the interest of keeping the proof short, we will only give the proof for the second possibility to compute T_{ij}; the proof for the first possibility follows along similar lines. First, we transform the expression of $\overline{\overline{T}}_{ij}^{(N)}$. Let η_1 and η_2 be the first and second term, respectively, of the equation at the top of the page. Modifying η_1 we obtain the second equation at the top of the page. Similarly, for η_2 we get the third equation at the top of the page. Let $\check{\boldsymbol{b}}$ be a (typical) input sequence and $\check{\boldsymbol{y}}$ be a corresponding (jointly typical) output sequence. Then the approximation $\overline{\check{\overline{T}}}_{ij}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}})$ of $\overline{\overline{T}}_{ij}^{(N)} = \eta_1 + \eta_2$ is given in the fourth equation at the top of the page, for finite N, and we have equality with probability 1 for $N \to \infty$. We omit the details of this last step; they essentially use the same results that were also used in [19]. A similar derivation leads to the corresponding expression for $\overline{\check{T}}_{i}^{(N)}(\check{\boldsymbol{b}}, \check{\boldsymbol{y}})$.

ACKNOWLEDGMENT

The authors gratefully acknowledge the reviewers' constructive comments. They are also grateful to Henry Pfister for pointing out to them the papers by Mevel *et al.* Moreover, P.O.Vontobel would like to thank Amos Lapidoth for his inspiring lectures at ETH Zurich and A. Kavčić would like to thank Xiao Ma for invaluable discussions during his stay at Harvard University.

REFERENCES

[1] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.

[2] H. K. Thapar and A. M. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magn.*, vol. MAG–23, no. 5, pp. 3666–3668, Sep. 1987.

[3] G. D. Forney, Jr., "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. IT–18, no. 3, pp. 363–378, May 1972.

[4] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott channel," *IEEE Trans. Inf. Theory*, vol. IT–35, no. 6, pp. 1277–1290, Nov. 1989.

[5] A. J. Goldsmith and P. P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 868–886, May 1996.

[6] A. Kavčić and A. Patapoutian, "A signal-dependent autoregressive channel model," *IEEE Trans. Magn.*, vol. 35, no. 5, pp. 2316–2318, Sep. 1999.

[7] D. Arnold, A. Kavčić, R. Kötter, H.-A. Loeliger, and P. O. Vontobel, "The binary jitter channel: A new model for magnetic recording," in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, Jun. 2000, p. 433.

[8] E. Zehavi and J. K. Wolf, "On runlength codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 1, pp. 45–54, Jan. 1988.

[9] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, Jul./Oct. 1948.

[10] S. Arimoto, "An algorithm for computing the capacity of arbitrary memoryless channels," *IEEE Trans. Inf. Theory*, vol. IT–18, no. 1, pp. 14–20, Jan. 1972.

[11] R. E. Blahut, "Computation of channel capacity and rate distortion functions," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 4, pp. 460–473, Jul. 1972.

[12] A. Kavčić, "On the capacity of Markov sources over noisy channels," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 2001, pp. 2997–3001.

[13] J. Lafferty and L. Wasserman, "Iterative Markov chain Monte Carlo computation of reference priors and minimax risk," in *Uncertainty in Artificial Intelligence: Proc. 17th Conf. (UAI-2001)*, San Francisco, CA, 2001, pp. 293–300.

[14] J. L. Holsinger, "Digital Communication over Fixed Time-Continuous Channels with Memory – With Special Application to Telephone Channels," Lab. Electron., MIT and Lincoln Lab., Cambridge, MA, 1964, Tech. Rep. 430 and Tech. Rep. 366.

[15] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991, Wiley Series in Telecommunications.

[17] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2260–2299, Oct. 1998.

[18] W. Hirt, "Capacity and Information Rates of Discrete-Time Channels with Memory," Ph.D. dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1988.

[19] D. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavčić, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug. 2006.

[20] S. Shamai (Shitz) and Y. Kofman, "On the capacity of binary and Gaussian channels with run-length limited inputs," *IEEE Trans. Commun.*, vol. 38, no. 5, pp. 584–594, May 1990.

[21] S. Shamai (Shitz), L. H. Ozarow, and A. D. Wyner, "Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. Inf. Theory*, vol. 37, no. 6, pp. 1527–1539, Nov. 1991.

[22] S. Shamai (Shitz) and S. Verdú, "Worst-case power-constrained noise for binary-input channels," *IEEE Trans. Inf. Theory*, vol. 38, no. 5, pp. 1494–1511, Sep. 1992.

[23] S. Shamai (Shitz) and R. Laroia, "The intersymbol interference channel: Lower bounds on capacity and channel precoding loss," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1388–1404, Sep. 1996.

[24] A. S. Khayrallah and D. L. Neuhoff, "Coding for channels with cost constraints," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 854–867, May 1996.

[25] B. Marcus, K. Petersen, and S. Williams, "Transmission rates and factors of Markov chains," *Contemp. Math.*, vol. 26, pp. 279–293, 1984.

[26] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," in *Proc. IEEE Int. Conf. Communications*, Helsinki, Finland, Jun. 2001, pp. 2692–2695.

[27] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to Markov channels," in *Proc. IEEE Int. Symp. Information Theory*, Washington, DC, Jun. 2001, p. 283.

[28] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite-state ISI channels," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 2001, pp. 2992–2996.

[29] P. O. Vontobel, "A Generalized Blahut-Arimoto Algorithm," Lab. Signal and Information Processing, 2002, Internal Rep. INT200203.

[30] P. O. Vontobel, "A generalized Blahut-Arimoto algorithm," in *Proc. IEEE Intern. Symp. Information Theory*, Pacifico Yokohama, Japan, Jun./Jul. 2003, p. 53.

[31] P. O. Vontobel and D. M. Arnold, "An upper bound on the capacity of channels with memory and constraint input," in *Proc. IEEE Information Theory Workshop*, Cairns, Australia, Sep. 2001, pp. 147–149.

[32] S. Yang, A. Kavčić, and S. Tatikonda, "Feedback capacity of finite-state machine channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 799–810, Mar. 2005.

[33] J. Chen and P. H. Siegel, "Markov processes asymptotically achieve the capacity of finite state intersymbol interference channels," *IEEE Trans. Inf. Theory*, 54, no. 3, pp. 1295–1303, Mar. 2008.

[34] C. E. Shannon, "Some geometrical results in channel capacity," *Fachberichte Verband Deutscher Elektrotechniker*, vol. 19, no. 2, pp. 13–15, 1956.

[35] C. E. Shannon, "Geometrische Deutung einiger Ergebnisse bei der Berechnung der Kanal Capazität [Geometrical meaning of some results in the calculation of channel capacity," *Nachrichtentech. Z. (N.T.Z.)*, vol. 10, no. 1, pp. 1–4, 1957.

[36] J. A. O'Sullivan, "Alternating minimization algorithms: From Blahut-Arimoto to expectation-maximization," in *Codes, Curves, and Signals*, A. Vardy, Ed. Norwell, MA: Kluwer Academic, 1998, pp. 173–192.

[37] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *J. Comp. Graph. Stat.*, vol. 9, no. 1, pp. 1–20, Mar. 2000.

[38] R. W. Yeung, *A First Course in Information Theory*. New York: Kluwer Academic/Plenum, 2002.

[39] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[40] G. D. Forney, Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[41] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[42] F. Le Gland and L. Mevel, "Basic properties of the projective product with applications to products of column-allowable nonnegative matrices," *Math. Control Signals Syst.*, vol. 13, pp. 41–62, 2000.

[43] F. Le Gland and L. Mevel, "Exponential forgetting and geometric ergodicity in Hidden Markov models," *Math. Control Signals Syst.*, vol. 13, pp. 63–93, 2000.

[44] L. Mevel and L. Finesso, "Asymptotical statistics of misspecified hidden Markov models," *IEEE Trans. Autom. Control*, vol. 7, no. 7, pp. 1123–1132, Jul. 2004.

[45] J. Yedidia, "An idiosyncratic journey beyond mean field theory," *Advanced Mean Field Methods, Theory and Practice*, pp. 21–36, Jan. 2001.

[46] J. Justesen and T. Høholdt, "Maxentropic Markov chains," *IEEE Trans. Inf. Theory*, vol. IT–30, no. 4, pp. 665–667, Jul. 1984.

[47] A. Kavčić, X. Ma, M. Mitzenmacher, and N. Varnica, "Capacity approaching signal constellations for channels with memory," in *Proc. 39th Allerton Conf. Communications, Control, and Computing*, Monticello, IL, Oct. 2001, pp. 311–320.

[48] A. Kavčić, X. Ma, and N. Varnica, "Matched information rate codes for partial response channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 973–989, Mar. 2005.

[49] J. B. Soriaga, H. D. Pfister, and P. H. Siegel, "On the low-rate Shannon limit for binary intersymbol interference channels," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 1962–1964, Dec. 2003.

[50] S. Yang and A. Kavčić, "Markov sources achieve the feedback capacity of finite-state machine channels," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 361.

[51] H. D. Pfister, "The derivatives of entropy rate and capacity for finite-state channels," Talk at BIRS Workshop on "Entropy of Hidden Markov Processes and Connections to Dynamical Systems. Banff, AB, Canada, Oct. 2007.

[52] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT–20, no. 2, pp. 284–287, Mar. 1974.

[53] H.-A. Loeliger, "A posteriori probabilities and performance evaluation of trellis codes," in *Proc. IEEE Int. Symp. Information Theory*, Trondheim, Norway, Jun./Jul. 1994, p. 335.

[54] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1994.