Energy-Optimized Lossless Compression: Rate-Variability Tradeoff*

Yihong Wu†
Princeton University
Princeton, NJ 08544, USA
Email: yihongwu@princeton.edu

Erik Ordentlich
Hewlett-Packard Laboratories
Palo Alto, CA 94304
Email: erik.ordentlich@hp.com

Marcelo J. Weinberger
Hewlett-Packard Laboratories
Palo Alto, CA 94304
Email: marcelo.weinberger@hp.com

***Abstract*—We pose the problem of energy-optimized lossless compression and analyze a simple compression framework in which energy consumption is given by a weighted sum of two components, respectively proportional to the compression rate and to the average number of bit flips that occur in a certain hardware register. The latter component, which we term variability, is meant to serve as a proxy for the energy consumption of the computations underlying the compression step. Our results include bounds on the rate-variability tradeoff for symbol-wise compression of discrete memoryless sources and a characterization of the asymptotically optimum tradeoff between rate and variability for block-wise compression.**

I. Introduction

In a wide range of data acquisition systems, the goal of lossless compression is to reduce transmission or storage energy. However, since better compression is obtained at the expense of more complicated encoding operations, optimizing for compression alone may degrade the overall energy efficiency of the system. Indeed, in [1], [2] it is shown through extensive experiments on an embedded processor that with several typical compression tools (LZ77, LZW, BWT, PPM), there is a net energy *increase* when compression is applied before transmission relative to the transmission of uncompressed data, as a result of computation. These works conclude that optimizing compression efficiency alone is not necessarily the best approach to minimizing total energy consumption. This conclusion applies even more when the goal is to reduce storage energy, since the relative weight of computation is larger in this case. This motivates the problem of designing efficient compressors to minimize total energy consumption. Such *energy-optimized* compressors could prove useful for embedded systems with severely limited resources, e.g., sensor networks and cyber-physical systems.

As a first step toward a rigorous study of energy-optimized lossless compression, we consider a simple framework for lossless compression in which code dependent energy consumption is predominantly comprised of two components. The first component is proportional to the *compression rate* (e.g., storage, transmission, etc.), while the second is proportional to the *code variability*, i.e., the average number of bit flips required to write a new codeword to a register. This energy model corresponds to an encoder based on a lookup table as it takes into account the switching activities in the digital system induced by the dynamics of encoding operations. It is motivated by the fact that the power consumption of CMOS circuits consists of *static* and *dynamic* power dissipation, where the latter dominates [3] and is proportional to the *transition frequency* of the logical signal. Excessive switching activity may cause larger peak and average power dissipation [4]. This model has been studied in coding for low-power I/O [5], where switching on the data bus is the main contributor to the power dissipation. Note that our energy consumption model does not include decompression and is, therefore, of primary relevance to systems in which decompression occurs infrequently or is carried out in a separate platform without resource constraints. Clearly, there are scenarios in which decompression energy consumption is important, but we leave its study for future work.

In the above framework, we model the total average energy expenditure of the compressor as a weighted sum of average codeword length and code variability. To gain insight into designing energy-optimized compressors, it is a fundamental problem to determine the optimal tradeoff between compression rate and variability. The main contribution of this paper consists of non-asymptotic bounds on the achievable rate-variability region that are asymptotically tight in the memoryless case. The performance of practical codes is also analyzed and compared to the optimal frontier. To simplify the presentation, we focus on prefix codes. However, the same conclusions hold for all uniquely decodable codes [6, p. 92] that are not necessarily prefix-free, as well as for near-lossless fixed-length codes.

In contrast to traditional lossless compression problems which focus on codeword length assignment, code variability depends on individual binary codewords. In fact, there can exist two prefix codes with the same pointwise codeword lengths but different variability. Another difficulty of our problem comes from the non-linear effect of code concatenation on variability: the codeword length of the concatenation of two codes is the sum of the constituent codeword lengths, while variability does not follow this linear relationship.

A related problem, known as *malleable coding*, is proposed in [7], which also concerns minimizing not only the average

* This research was funded in part by the US Government's Nano-Enabled Technology Initiative.

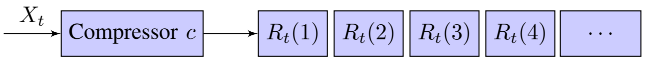
† The work of Yihong Wu was done while visiting Hewlett-Packard Laboratories, Palo Alto, CA.

codeword length, but also the cost when changing the representation to match an updated message with respect to a general edit cost. The solution of the malleable coding problem is left open in [7]. Our result provides an explicit solution under the restrictions of identical fixed-length component codes, Hamming distance, and a decoupled (independent) modification channel.

In our analysis of the optimal tradeoff between compression rate and variability in Section IV, we will allow for the possibility of compressing symbols in blocks of length n. The achievability result will require n to grow, which, in our simple compression framework (as explained in more detail in the next section), would translate into an exponentially growing lookup table. Thus, for sufficiently large n, the exponentially growing energy expended on table lookups, as dictated by, e.g., growing wire lengths, which is not included in our model, may dominate the energy consumption corresponding to rate and variability. In view of this observation, the achievability analysis is relevant primarily as a contribution to the malleable coding setting of [7], where variability may also model other costs of data storage such as device wear, of significance to flash-based and emerging non-volatile storage technologies. The converse result, on the other hand, which holds for all block lengths n, is very much relevant also to the energy optimization problem as well.

II. Setup

Consider a lossless data compression framework that works according to the following steps:



1) The compressor fetches a data symbol from the incoming data stream;
2) Using the binary representation of the symbol as the address, a codeword is read from the codebook stored in a lookup table;
3) The compressor writes the codeword to a binary register from left to write, without overwriting bits beyond the end of the codeword;
4) The content of the register, up through the end of the last codeword, is stored to non-volatile memory or transmitted.

The energy consumption of the first step is independent of the coding method. The cost of the second and fourth step is affine in the codeword length, with the addressing step being largely independent of the code, while the cost of the third step is proportional to the code variability, which is the average number of bit flips required to update the register. Therefore, we can model the expected code-dependent portion of the total energy consumption as a weighted sum of code length and variability.

To formulate the problem mathematically, let X be a random variable taking values in a finite alphabet $\mathcal{X}$. Let $c : \mathcal{X} \to \{0,1\}^*$ be a prefix code and $C = c(X)$. Denote by $L = \ell(C)$ the codeword length and $\ell_{\max}$ the maximal length of codewords. Let $C(i)$ be the i^{th} bit of C if $i \le L$; otherwise $C(i) = \Lambda$ (empty symbol), so that $C(i)$ can take on three values. The *average codeword length* of c can be written as

$$\begin{aligned} \mathsf{L} = \mathbb{E}[L] &= \sum_{i\ge 1} \mathbb{P}\{L \ge i\} && (1)\\ &= \sum_{i\ge 1} (\mathbb{P}\{C(i)=0\} + \mathbb{P}\{C(i)=1\}). && (2) \end{aligned}$$

Next, we define the *code variability*. Consider a discrete memoryless source (DMS) $\{X_t\}$ with marginal distribution P_X. Denote $C_t = c(X_t)$. Let $R_t(i)$ be the i^{th} bit of the register at time t, with $R_0(i) = 0$ for each i. According to our framework, for $t > 0$, the contents of the register satisfy the following Markov dynamics:

$$R_t(i) = \begin{cases} R_{t-1}(i) & \ell(C_t) < i \\ C_t(i) & \ell(C_t) \ge i, \end{cases} \tag{3}$$

with transition probability from 0 to 1 given by:

$$\begin{aligned} &\mathbb{P}\{R_t(i) = 1 | R_{t-1}(i) = 0\} \\ &= \mathbb{P}\{\ell(C_t) \ge i, C_t(i) = 1 | R_{t-1}(i) = 0\} && (4)\\ &= \mathbb{P}\{\ell(C_t) \ge i, C_t(i) = 1\} && (5)\\ &= \mathbb{P}\{C(i) = 1\}\,. && (6) \end{aligned}$$

Similarly, the transition probability from 1 to 0 is $\mathbb{P}\{C(i) = 0\}$. Therefore, assuming state probabilities are row vectors, for each i, the Markov chain $\{R_t(i) : t \ge 1\}$ is time-homogeneous with transition matrix

$$\begin{pmatrix} \mathbb{P}\{C(i) \ne 1\} & \mathbb{P}\{C(i) = 1\} \\ \mathbb{P}\{C(i) = 0\} & \mathbb{P}\{C(i) \ne 0\} \end{pmatrix}. \tag{7}$$

Hence, the stationary distribution $\pi(i) = (\pi(i)_0, \pi(i)_1)$ exists, and is given by

$$\pi(i)_0 = \frac{\mathbb{P}\{C(i) = 0\}}{\mathbb{P}\{C(i) = 0\} + \mathbb{P}\{C(i) = 1\}}, \tag{8}$$

$$\pi(i)_1 = \frac{\mathbb{P}\{C(i) = 1\}}{\mathbb{P}\{C(i) = 0\} + \mathbb{P}\{C(i) = 1\}}. \tag{9}$$

Letting $R_t^\ell = [R_t(1), \ldots, R_t(\ell)]$, the *variability* of code c is defined as the number of bits we need to flip per encoding operation in the steady state:

$$\begin{aligned} \mathsf{V} &= \lim_{t\to\infty} \mathbb{E}\left[d_{\mathsf{H}}(R_t^{\ell(C_{t+1})}, C_{t+1}) \right] && (10)\\ &= \sum_{i\ge 1} (\pi(i)_0 \mathbb{P}\{C(i)=1\} + \pi(i)_1 \mathbb{P}\{C(i)=0\}) && (11)\\ &= \sum_{i\ge 1} \frac{2\,\mathbb{P}\{C(i)=0\}\mathbb{P}\{C(i)=1\}}{\mathbb{P}\{C(i)=0\} + \mathbb{P}\{C(i)=1\}}, && (12) \end{aligned}$$

where d_{H} denotes the Hamming distance and the last equality follows from (8) – (9). The memory of the rewrite process is captured by the stationary distribution. From (2) and (12), it is interesting to notice that V and L only depend on the *marginal* distribution of the compressor output bits. In the special case of fixed-length codes, the variability is simply the expected Hamming distance between two *independent* codewords.

III. Bounds on the achievable (L,V)-region

For a given random variable X, the optimization of energy consumption for our simple framework and energy model will depend on the region of achievable (L,V)-pairs of all prefix codes. The following upper bound follows directly from applying the inequality $\sqrt{uv} \le \frac{u+v}{2}$ to (2) and (12).

Theorem 1. *For any code,*

$$\frac{\mathsf{V}}{\mathsf{L}} \le \frac{1}{2}, \tag{13}$$

with equality if and only if, for each i,

$$\mathbb{P}\{C(i)=1\} = \mathbb{P}\{C(i)=0\}. \tag{14}$$

To establish a lower bound on V we use the following lemma.

Lemma 1.

$$H(P) \le \sum_{i\ge 1} \mathbb{P}\{L \ge i\} h(\mathbb{P}\{C_i = 1 | L \ge i\}). \tag{15}$$

Proof: This result can be proved via induction on the maximal codeword length. Instead, we give an operational proof by constructing a lossless compression scheme for the DMS P, whose average compression rate is the right-hand side of (15), hence not exceeding $H(P)$. In this scheme, we first apply the compressor c to $X_1,\dots,X_n$ independently sampled from P, to obtain variable-length strings $C_1,\dots,C_n$. Then, for each i, $1 \le i \le \ell_{\max}$, we form a binary word B_i consisting of $\{C_t(i) : L_t \ge i\}$ ordered in increasing t, and apply a Huffman code to B_i. Clearly, $C_1,\dots,C_n$ can be reconstructed from $\{B_i, 1 \le i \le \ell_{\max}\}$ since, due to the prefix property of the underlying code c, the unspecified locations i in which $C_i = \Lambda$ are implicitly available. Since the average code length is no more than $H(B_i)+1$ bits, and $H(B_i) = n\mathbb{P}\{L \ge i\} h(P\{C(i) = 1 | L \ge i\})$, the expected normalized total code length is upper bounded by $\ell_{\max}/n + \sum_i \mathbb{P}\{L \ge i\} h(P\{C(i) = 1 | L \ge i\})$, the right-hand side of (15) plus an arbitrarily small term. ■

We also define the functions $v : [0,1] \to [0,\frac{1}{2}]$ and $w : [0,1] \to [0,\frac{1}{2}]$ by

$$v(p) = 2p(1-p), \tag{16}$$

$$w(z) = v(h^{-1}(z)), \tag{17}$$

where $h^{-1} : [0,1] \to [0,\frac{1}{2}]$ is the inverse of the binary entropy function h on $[0,\frac{1}{2}]$. Lemma 2 below, whose proof is omitted, states properties of these functions.

Lemma 2. *1) w is a strictly increasing and strictly convex function, with $w(0) = 0$ and $w(1) = \frac{1}{2}$.*

2) For any $a > 0$, the function $u_a : [a,\infty) \to (0,\frac{a}{2}]$ defined by $u_a(x) = x \cdot w\left(\frac{a}{x}\right)$ is strictly decreasing and strictly convex, with $u_a(a) = \frac{a}{2}$.

In view of (12), (16), and the equality between the addends of (1) and (2), V takes the form

$$\mathsf{V} = \sum_{i\ge 1} \mathbb{P}\{L \ge i\} v(\mathbb{P}\{C(i) = 1 | L \ge i\}). \tag{18}$$

Theorem 2. *For any prefix code,*

$$\frac{\mathsf{V}}{\mathsf{L}} \ge w\left(\frac{H(P)}{\mathsf{L}}\right), \tag{19}$$

where $\mathsf{L} \ge H(P)$.

Proof: Using (18),

$$\begin{aligned}
\frac{\mathsf{V}}{\mathsf{L}} &= \sum_{i\ge 1} \frac{\mathbb{P}\{L \ge i\}}{\mathsf{L}} v(\mathbb{P}\{C(i) = 1 | L \ge i\}) && (20)\\
&= \sum_{i\ge 1} \frac{\mathbb{P}\{L \ge i\}}{\mathsf{L}} w(h(\mathbb{P}\{C(i) = 1 | L \ge i\})) && (21)\\
&\ge w\left(\frac{\sum_{i\ge 1} \mathbb{P}\{L \ge i\} h(\mathbb{P}\{C(i) = 1 | L \ge i\})}{\mathsf{L}}\right) && (22)\\
&\ge w\left(\frac{H(P)}{\mathsf{L}}\right), && (23)
\end{aligned}$$

where (22) is due to the convexity of w and (1), while (23) follows from the monotonicity of w and Lemma 1. ■

We note that Theorems 1 and 2 together imply that any code that exactly attains the source entropy will necessarily have variability equal to half of the entropy.

IV. Achievable (L,V)-regions for DMS

In this section, we allow for the possibility of compressing symbols in blocks of length n and determine the optimal asymptotic achievable tradeoff between L and V. We consider the following two types of achievable regions. The first type consists of operating points of all prefix codes.

Definition 1. Let X^n be independently sampled from P. The collection of all operating points of prefix codes for X^n is defined as

$$\mathcal{R}^{(n)}(P) = \left\{ \left(\frac{\mathsf{L}(c)}{n}, \frac{\mathsf{V}(c)}{n}\right) : c : \mathcal{X}^n \to \{0,1\}^* \text{ is a prefix code} \right\}. \tag{24}$$

The collection of operating points of all prefix codes with different blocklength is defined as

$$\mathcal{R}^*(P) = \mathbf{cl} \bigcup_{n\ge 1} \mathcal{R}^{(n)}(P). \tag{25}$$

The second type involves the limit in the blocklength.

Definition 2. A pair $(\tilde{\mathsf{L}}, \tilde{\mathsf{V}})$ is said to be achievable for P if there exists a sequence of prefix codes $c_n : \mathcal{X}^n \to \{0,1\}^*$, such that

$$\limsup_{n\to\infty} \frac{\mathsf{L}(c_n)}{n} = \tilde{\mathsf{L}}, \quad \limsup_{n\to\infty} \frac{\mathsf{V}(c_n)}{n} = \tilde{\mathsf{V}}. \tag{26}$$

The achievable region $\mathcal{R}(P)$ is the collection of all achievable pairs for P.

By Definitions 1 and 2, we have

$$\mathcal{R}(P) = \mathbf{cl} \limsup_{n\to\infty} \mathcal{R}^{(n)}(P) \tag{27}$$

$$= \mathbf{cl} \bigcap_{N\geq 1} \bigcup_{n \geq N} \mathcal{R}^{(n)}(P) \subseteq \mathcal{R}^*(P). \tag{28}$$

Notice that even though a code on blocks of length i defines a code on blocks of length $2i$ by concatenation, the sequence of regions $\mathcal{R}^{(n)}(P)$ is not necessarily nested, due to the complicated dynamics of code concatenation and its non-linear effect on the variability. Indeed, while the codeword length of the concatenation of two codes is the sum of individual code lengths, the variability is in general strictly larger than the sum of individual code variabilities. This property follows from the fact that the dynamics of two constituent codes are intertwined when concatenated. Thus, it is surprising that, as the next theorem shows, $\mathcal{R}^*(P)$ and $\mathcal{R}(P)$ are in fact equal.

Theorem 3.

$$\mathcal{R}^*(P) = \mathcal{R}(P) \tag{29}$$

$$= \left\{ (l, v) : l \geq H(P), \quad w\left(\frac{H(P)}{l}\right) \leq \frac{v}{l} \leq \frac{1}{2} \right\}, \tag{30}$$

where w is defined in (17).

If we normalize both L and V by the source entropy $H(P)$, denoting the normalized pair by $(\bar{\mathsf{L}}, \bar{\mathsf{V}})$, the achievable region is independent of the source statistics (see Fig. 1):

$$\bar{\mathcal{R}} = \left\{ (\bar{l}, \bar{v}) : \bar{l} \geq 1, \quad w\left(\frac{1}{\bar{l}}\right) \leq \frac{\bar{v}}{\bar{l}} \leq \frac{1}{2} \right\}. \tag{31}$$

Therefore, changing the source distribution only scales the achievable region in Fig. 1 but does not alter its shape. Note that the vertex $(1, \frac{1}{2})$ corresponds to optimal compression. In (31) the “efficient frontier”, corresponding to the lower boundary, is of primary interest for optimal codes, although the upper bound can help assess how much better a particular code is relative to the worst possible tradeoff. Notice that the shape of this boundary follows from the second part of Lemma 2.

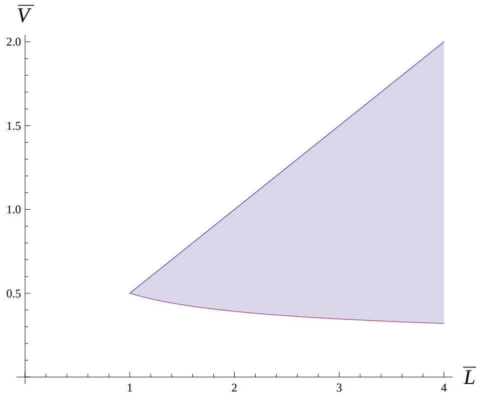


Fig. 1. Normalized achievable region $\bar{\mathcal{R}}$.

Proof of Theorem 3: The converse part of the proof follows from Theorems 1 and 2.

We shall prove the achievability of the lower boundary. For any $\epsilon > 0$, let

$$T_n^\epsilon \triangleq \{x^n : |\hat{H}(x^n) - H(P)| \leq \epsilon\}, \tag{32}$$

where $\hat{H}(x^n)$ denotes the empirical entropy of x^n. Let M_n^ϵ denote the collection of types P of sequences in T_n^ϵ and T_{P} the set of sequences x^n with type P. Thus, $T_n^\epsilon = \bigcup_{\mathsf{P}\in M_n^\epsilon} T_{\mathsf{P}}$ and $|M_n^\epsilon|$ is upper bounded by $(n+1)^{|\mathcal{X}|}$. Moreover, $\mathbb{P}\{X^n \in T_n^\epsilon\} \to 1$ and, by [8, Chapter 1, Lemma 2.3] and (32), for any $\mathsf{P} \in M_n^\epsilon$ and sufficiently large n,

$$n(H(P) - 2\epsilon) \leq \log |T_{\mathsf{P}}| \leq n(H(P) + \epsilon) \tag{33}$$

where logarithms are taken to base 2. Clearly, the lower boundary can be parametrized as

$$\left\{ \left(\frac{H(P)}{h(\alpha)}, \frac{H(P)v(\alpha)}{h(\alpha)} \right) : 0 < \alpha \leq \frac{1}{2} \right\}. \tag{34}$$

For any $0 < \alpha \leq \frac{1}{2}$, let

$$L'(\alpha) = \left(\frac{H(P) + 2\epsilon}{h(\alpha)} \right) n. \tag{35}$$

We prove the existence of a prefix code $c_\alpha : \mathcal{X}^n \to \{0,1\}^*$ such that

$$\mathsf{L}(c_\alpha) \;= L'(\alpha) + o(n) \tag{36}$$

$$\mathsf{V}(c_\alpha) \;\leq v(o(1) + \alpha)L'(\alpha) + o(n). \tag{37}$$

To this end, we encode T_n^ϵ as follows, so that the marginal distribution of codeword bits is close to α:[1]

Step 1: If $x^n \notin T_n^\epsilon$, encode it with ‘0’ followed by an uncompressed representation of x^n using $n \log |\mathcal{X}|$ bits.

Step 2: If $x^n \in T_{\mathsf{P}}$ with $\mathsf{P} \in M_n^\epsilon$, encode it with a ’1’ followed by x^n encoded according to the following constant weight code. Let

$$\ell(\alpha) = h^{-1}((1-\delta)h(\alpha))L'(\alpha) \tag{38}$$

where $\delta{=}\epsilon(1-\epsilon)/(H(P)+2\epsilon)$. Clearly, $\ell(\alpha)/L'(\alpha) < \alpha$, and it is easy to see, by (33) and (35), that the set $\mathcal{L}'$ of $L'(\alpha)$-bit binary vectors with Hamming weight $\ell(\alpha)$ is, for sufficiently large n, larger than $|T_{\mathsf{P}}|$. The constant weight code of interest is then obtained by injectively mapping T_{P} into $\mathcal{L}'$ in such a way that all vectors in $\mathcal{L}'$ that are cyclic shifts of a mapped vector are included in the image before other vectors. The resulting code is thus nearly a cyclic code, in the sense that with the exception of an exponentially negligible fraction of at most $L'(\alpha) - 1$ codewords, all cyclic shifts of a codeword are also codewords. Following the constant weight encoding, encode the type of x^n using a fixed-length code of $|\mathcal{X}| \log(n+1) = o(n)$ bits. Then append it to the codeword obtained above.

Note that for $x^n \in T_n^\epsilon$, the codeword length is fixed at $L'(\alpha) + o(n)$. Since $\mathbb{P}\{X^n \notin T_n^\epsilon\} \to 0$, the above code c_α encodes X^n losslessly with average length (36). Next,

[1] For simplicity, we loosely avoid truncating various quantities to integer values. The derivations can be made completely rigorous with respect to this issue.

we analyze its normalized variability. First, we bound the conditional marginal distribution of the codeword bits $C(i)$, $2 \leq i \leq L'(\alpha)+1$. Since $\mathbb{P}\{L \geq i\} \geq \mathbb{P}\{X^n \in T_n^\epsilon\} = 1 - o(1)$,

$$\mathbb{P}\{C(i) = 1 | L \geq i\} = \frac{\mathbb{P}\{C(i) = 1\}}{\mathbb{P}\{L \geq i\}} \tag{39}$$

$$= \mathbb{P}\{C(i) = 1\} + o(1) \tag{40}$$

and

$$\mathbb{P}\{C(i) = 1\}$$

$$= \mathbb{P}\{C(i) = 1, X^n \notin T_n^\epsilon\} + \mathbb{P}\{C(i) = 1, X^n \in T_n^\epsilon\} \tag{41}$$

$$\leq \sum_{\mathsf{P} \in M_n^\epsilon} \mathbb{P}\{C(i) = 1 | X^n \in T_{\mathsf{P}}\} \mathbb{P}\{X^n \in T_{\mathsf{P}}\} + o(1) \tag{42}$$

$$= \sum_{\mathsf{P} \in M_n^\epsilon} \left(\frac{\ell(\alpha)}{L'(\alpha)} + o(1) \right) \mathbb{P}\{X^n \in T_{\mathsf{P}}\} + o(1) \tag{43}$$

$$\leq \alpha \mathbb{P}\{X^n \in T_n^\epsilon\} + o(1) \leq \alpha + o(1) \tag{44}$$

where (43) follows from the fact that all sequences of a given type have the same probability and are thus conditionally uniformly distributed among each cyclic shift of each codeword with constant weight $\ell(\alpha)$ (with the small exceptional set noted above contributing the new $o(1)$ term, which is independent of i), and (44) follows from the fact noted above that $\ell(\alpha)/L'(\alpha) < \alpha$.

Note that for $i > L'(\alpha)+1+|\mathcal{X}|\log(n+1)$, $\mathbb{P}\{L \geq i\} = o(1)$ and for $i > \max\{L'(\alpha)+1+|\mathcal{X}|\log(n+1), 1+n\log|\mathcal{X}|\}$, $\mathbb{P}\{L \geq i\} = 0$. Since v is increasing in $[0, \frac{1}{2}]$, by (18) and the preceding considerations, we obtain (37), which implies the achievability of $\left(\frac{H(P)+2\epsilon}{h(\alpha)} + o(1), \frac{(H(P)+2\epsilon)v(o(1)+\alpha)}{h(\alpha)} + o(1)\right)$. By the arbitrariness of $\epsilon > 0$, the continuity of $v(\cdot)$, the parametrization (34), and $\alpha \in (0, \frac{1}{2}]$, the lower boundary is achievable. ■

V. Minimizing the total energy consumption

Minimizing the total energy consumption over all possible prefix codes, we define

$$E(\lambda) \triangleq \min \quad \{\lambda v + l : (l, v) \in \mathcal{R}^*(P)\} \tag{45}$$

where λ is the cost of bit flipping relative to the average cost of codeword length. Therefore, $E(\lambda)$ is simply the Fenchel-Legendre transform of the (strictly convex) lower boundary of region $\mathcal{R}^*(P)$ in (30), namely

$$E(\lambda) = H(P) \min_{\bar{l} \geq 1} \lambda \, \bar{l} \, w\left(\frac{1}{\bar{l}}\right) + \bar{l}. \tag{46}$$

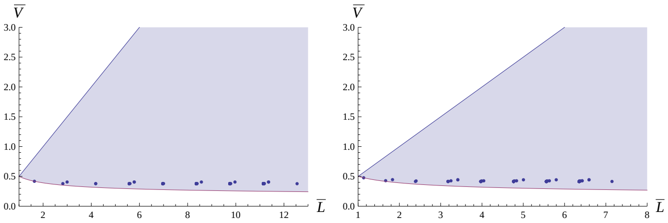
Interestingly, there is a critical slope at $\bar{l} = 1$:

$$\left. \frac{\mathrm{d}\left[\bar{l} \cdot w\left(\frac{1}{\bar{l}}\right)\right]}{\mathrm{d}\bar{l}} \right|_{\bar{l}=1} = \frac{1}{2} - \log_e 2 \approx -0.193, \tag{47}$$

which is the most negative slope on the lower boundary. This implies that the minimizer of (45) is the vertex $(1, 1/2)$ for all $\lambda \leq \frac{2}{2\log_e 2 - 1} \approx 5.17$. Therefore, $E(\lambda) = \left(\frac{\lambda}{2} + 1\right) H(P)$ for all $\lambda \leq 5.17$, that is, if the cost of variability is sufficiently cheap (i.e., cheaper than 5.17 times the cost of codeword length), the best strategy is to minimize compression rate.

VI. Performance of practical codes

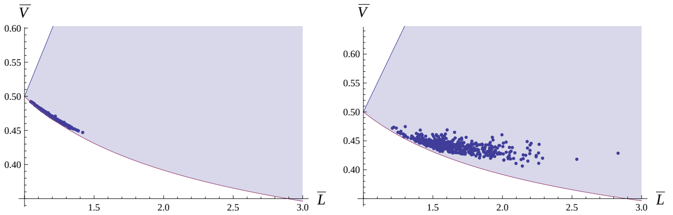
We apply the family of Golomb codes [9] with parameter M ranging from 1 to 256 to truncated geometric distributions and plot its rate-variability tradeoff in Fig. 2. Numerical results suggest that Golomb codes operate close to the boundary.



(a) $p = 0.15$ (b) $p = 0.3$

Fig. 2. Operating points of Golomb codes ($M = 1, \ldots, 256$) with respect to truncated geometric distribution $P = [1-p, (1-p)p, \ldots, (1-p)p^{N-2}, p^{N-1}]$ and $N = 256$.

The unary code is a special case of Golomb codes ($M = 1$). As a sanity check of Theorem 3, Fig. 3 plots the operating points of the unary code of length 8 for randomly generated sources. Of course, all points lie in the region given by Theorem 3. From numerical results, we see that the unary code operates fairly close to the optimal boundary if the source alphabet is sorted according to decreasing probabilities; otherwise, the gap to optimal performance is substantial.



(a) Sorted (b) Unsorted

Fig. 3. Operating points of unary code of length 8 for 500 instances of randomly generated sources.

References

[1] K. Barr and K. Asanović, "Energy aware lossless data compression," in *Proceedings of the First International Conference on Mobile Systems, Applications and Services (Mobisys)*, May 2003, pp. 244 – 259.

[2] K. C. Barr and K. Asanović, "Energy-aware lossless data compression," *ACM Transactions on Computer Systems*, vol. 24, no. 3, p. 291, 2006.

[3] K. Eshraghian and N. Weste, *Principles of CMOS VLSI design: a systems perspective*. Addison Wesley, 1994.

[4] M. Tehranipour, M. Nourani, K. Arabi, and A. Afzali-Kusha, "Mixed RL-Huffman encoding for power reduction and data compression in scan test," in *Proc. of ISCAS*, vol. 2, 2004, pp. 681–684.

[5] M. Stan and W. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 1, pp. 49–58, 1995.

[6] D. J. MacKay, *Information theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[7] L. Varshney, J. Kusuma, and V. Goyal, "Malleable coding with edit-distance cost," in *Proceedings of the 2009 IEEE International Symposium on Information Theory*, Seoul, South Korea, Jun. 2009.

[8] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, Inc., 1982.

[9] S.W. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory*, IT–12(3):399–401, Jul. 1966.