The Huffman Code Is Optimal, but Only Within the Constraint That Code Lengths Be Whole Numbers

— With probabilities $1/2$, $1/4$, $1/8$ and $1/8$ the average length 1.750000 meets the entropy 1.750000 exactly — the separator is whether each symbol's probability is a power of 2 — [Paper 752]

Yuuki YAMAGISHI (山岸 勇輝)
Yamagishi Complexity Torus Lab, Japan
September 2026

— *From thinking to syncing.* —

Abstract

The Huffman code is known as the code of shortest average length. That optimality does not mean reaching the entropy.Optimal within what, and how far short, is counted.No new theorem or law is claimed. First, with probabilities that are powers of 2, it arrives exactly——$1/2$, $1/4$, $1/8$, $1/8$: average length 1.750000, entropy 1.750000 (Section 2). **Second, for two skewed symbols, the optimal code is far off——at probability 0.99 the entropy is 0.080793 bits, while symbol-by-symbol Huffman spends 1.000000 bit** (Section 3). **Third, and this is the core. The "optimal code" spends 12.377289 times the entropy——because lengths must be whole numbers and no symbol can be given less than 1 bit** (Section 3). **Fourth, grouping approaches, but only from above——groups of 2, 4 and 8 give 0.514950, 0.272698 and 0.157158 bits, always within $1/k$** (Section 3). **Fifth, even at 8, still 1.945193 times——every Kraft sum is 1.000000, so the code has no gaps** (Section 3). **Sixth, the separator is whether each symbol's probability is a power of 2——if so, symbol-by-symbol coding arrives; if not, the gap shrinks with grouping but never vanishes** (Section 4). **Keywords**: Huffman code, entropy, code length, Kraft inequality, coding symbols in groups, whole-number constraint.

Scope of this paper (scope note): No new theorem or law is claimed——that the Huffman code minimises average length among prefix codes, that average length lies at least at the entropy and below entropy plus 1, that grouping brings the gap below $1/k$, and the Kraft inequality are all known (Huffman 1952, Cover and Thomas 2006).Codes without the whole-number constraint are not treated——schemes such as arithmetic coding, which give symbols fractional lengths, are only named.The source is limited——only memoryless two- and four-symbol sources with known probabilities are considered.Cost is not treated——the exponential growth of the code table with grouping, and delay, are not treated.No real data are claimed——particular files and compressors are not treated.Relation to earlier papers: Paper 304 showed that channel capacity is attained at no finite code length and that the loss falls only as $1/\sqrt{n}$ ——its scope note stated that which codes approach capacity would not be treated at all. This is the noiseless source side: the entropy is met exactly symbol by symbol when probabilities are powers of 2, and otherwise approached within $1/k$. Whether it is reached at finite length is decided by the form of the probabilities.Paper 163 is also cited—— it separated "a good code exists" from "here is a good code". Huffman is on the "here is" side, a construction shown optimal within whole-number lengths.What is added is confirming exact arrival for powers of 2, showing that for two skewed symbols the "optimal code" spends **12.377289** times the entropy, tabulating the gap, $1/k$ and the Kraft sum for each grouping, and placing the separator on whether probabilities are powers of 2.

1. Notation and the Question

Setting: let symbol i have probability p_i and a whole-number code length ℓ_i. The entropy H is the lower bound on average length, and the Huffman code minimises the average length L among prefix codes. Coding symbols in groups of k, the length per symbol satisfies the following.

$$H = -\sum_i p_i \log_2 p_i, \qquad \sum_i 2^{-\ell_i} \le 1, \qquad H \le \frac{L_k}{k} < H + \frac{1}{k}$$

The middle one is the Kraft inequality. Only when $\ell_i = -\log_2 p_i$ is a whole number can $L = H$.

Question: does the optimal code reach the entropy?

2. With Powers of 2, It Arrives Exactly

Computation 2.1: a Huffman code was built for four symbols with probabilities $1/2$, $1/4$, $1/8$ and $1/8$.

quantity	value
average length	1.750000
entropy	1.750000
difference	0

> **Observation 2.2**
>
> with probabilities that are powers of 2, the Huffman code meets the entropy exactly.The lengths 1, 2, 3, 3 are $-\log_2 p_i$ themselves——here the whole-number constraint binds nothing.

3. For Two Skewed Symbols, the Optimal Code Is Far Off

Computation 3.1: for two symbols with one at probability 0.99 (entropy 0.080793 bits), Huffman codes were built on groups of k symbols.

k	per symbol	gap to entropy	bound $1/k$	Kraft sum
1	1.000000	0.919207	1.000000	1.000000
2	0.514950	0.434157	0.500000	1.000000
4	0.272698	0.191904	0.250000	1.000000
8	0.157158	0.076365	0.125000	1.000000

> **Observation 3.2**
>
> this is the core: the symbol-by-symbol “optimal code” spends 12.377289 times the entropy.Two symbols need 1 bit each however assigned, so 1.000000 bit is paid for information that 0.080793 bits would carry ——Huffman is optimal, but within the constraint that lengths be whole numbers.

Observation 3.3

grouping approaches, but only from above.The gap shrinks, 0.434157, 0.191904, 0.076365, always below $1/k$, and at 8 symbols 1.945193 times the entropy remains——grouping only dilutes the fractional length inside larger symbols; the constraint itself is still there.

Observation 3.4

every Kraft sum is 1.000000.No branch of the code tree is unused——the waste comes not from gaps but from lengths being whole numbers.

Observation 3.5

the symbol-by-symbol gap 0.919207 is the same value as the capacity of the binary symmetric channel with error rate 0.01 that appears in Paper 163 and Paper 304.Both are the value of $1 - h(0.01)$; there it is a usable rate, here a length overpaid——the value of one expression, with different meanings.

4. The Separator

Observation 4.1

the separator is whether each symbol's probability is a power of 2.If so, symbol-by-symbol coding meets the entropy exactly; if not, grouping shrinks the gap below $1/k$ without removing it——"is Huffman optimal?" has no answer until one says optimal within what, and what form the probabilities take.

Observation 4.2

beside Paper 304.Paper 304 showed channel capacity attained at no finite code length and left aside which codes approach it——this is the source side: with powers of 2 the limit is reached at finite length, otherwise approached from above. Whether it is reached at finite length is decided by the form of the probabilities.

Observation 4.3

beside Paper 163 as well.Paper 163 separated “a good code exists” from “here is a good code”——Huffman is on the “here is” side, and its optimality carries the condition of whole-number lengths.

Observation 4.4

what is not said.Arithmetic coding, sources with memory, table size and real data are not treated——what is shown is what optimal means here, and how far it stands from the entropy.

5. Roll Call and Closing

1. **1.750000 with powers of 2** (Computation 2.1): **exact arrival.**
2. **entropy 0.080793** (Computation 3.1): **two skewed symbols.**
3. **symbol by symbol 1.000000** (Computation 3.1): **the whole-number constraint.**
4. **12.377289 times** (Observation 3.2): **the core.**
5. **groups of 2: 0.514950** (Computation 3.1): **approaching.**
6. **groups of 8: 0.157158** (Computation 3.1): **from above.**
7. **still 1.945193 times** (Observation 3.3): **not vanishing.**
8. **Kraft sum 1.000000** (Observation 3.4): **no gaps.**
9. **gap 0.919207** (Observation 3.5): **one expression, another meaning.**
10. **probabilities powers of 2 or not** (Observation 4.1): **the separator; Paper 304 and Paper 163.**

Closing: **The Huffman code is optimal, but only within the constraint that code lengths be whole numbers.With probabilities $1/2$, $1/4$, $1/8$ and $1/8$ the average length 1.750000 meets the entropy 1.750000 exactly——because the lengths are $-\log_2 p_i$ themselves.But for two symbols with one at probability 0.99, against an entropy of 0.080793 bits symbol-by-symbol Huffman spends 1.000000 bit, 12.377289 times the entropy——no symbol can be given less than 1 bit.Grouping 2, 4 and 8 symbols brings it to 0.514950, 0.272698 and 0.157158 bits, always within $1/k$, yet at 8 still 1.945193 times the entropy.The separator is whether each symbol’s probability is a power of 2——“is Huffman optimal?” has no answer until one says optimal within what, and what form the probabilities take.Placed among the earlier papers——Paper 304 showed channel capacity unreached at finite length; on the source side, whether the limit is reached at finite length depends on the probabilities. On the “here is” side of Paper 163 stands a conditional optimum.To be honest——arithmetic coding, sources with memory and table size are not treated.**

References

1. Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory* (2nd ed.). Wiley.
2. Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*.
3. Yamagishi, Y. (2026). *"A Good Code Exists" and "Here Is a Good Code" Are Different Sentences* [Paper 163]. Zenodo. 10.5281/zenodo.22092677
4. Yamagishi, Y. (2026). *"Capacity" Is a Number Attainable Only at Infinite Length* [Paper 304]. Zenodo. 10.5281/zenodo.22172081

— *From thinking to syncing.* —

Yuuki YAMAGISHI (山岸 勇輝)

© 2026 Yamagishi Complexity Torus Lab, Japan · Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)