解 説

準瞬時 FV 符号（AIFV 符号）
――ハフマン符号に勝る圧縮率を達成する符号――

AIFV Code to Achieve Better Compression Rate than the Huffman Code

山本博資

Abstract

ハフマン符号は，定常無記憶なデータ系列を最も効率良く圧縮できる最適なデータ圧縮符号としてよく知られている．しかし，2015 年に Yamamoto, Tsuchihashi, Honda によって提案された準瞬時 FV 符号（AIFV 符号：Almost Instantaneous Fixed-to-Variable length code）を用いると，ハフマン符号より更に良い圧縮率を実現することができる．AIFV 符号は 2 個の符号木から構成されるが，その AIFV 符号の符号木の構造，符号化／復号アルゴリズム，平均符号長，最適な AIFV 符号木の構成法などについて解説する．更に，m 個の符号木を用いる AIFV-m 符号をはじめとする様々な拡張符号や関連する研究について紹介する．

キーワード：AIFV 符号，ハフマン符号，AIFV-m 符号，AIVF 符号，データ圧縮，反復最適化

1. は じ め に

データ圧縮符号のうち，データ系列を一文字ごとに可変長の符号語に符号化する符号は，FV 符号（Fixed-to-Variable length code）と呼ばれる．ここでは，符号語が $\{0,1\}$ の二元系列である二元 FV 符号を考える．FV 符号のうち，任意のデータ系列を誤りなく復号できるものを一意復号可能符号という．一意復号可能符号のうち，各符号語の最終ビットを読み込んだ時点でその符号語の復号を完了できる符号を瞬時符号という．瞬時符号は，任意の符号語が他の全ての符号語の語頭と異なっているため，語頭符号（prefix code）とも呼ばれる[1]．

FV 符号は符号木を用いて表現することができる．情報源アルファベット $\mathcal{X}=\{a,b,c,d\}$ に対する符号木の例を図 1 に示す．符号木の根から文字 $x\in\mathcal{X}$ が割り振られた節点までのパスが，x の符号語を表している．瞬時符号（語頭符号）となるためには，全ての文字 x は葉に割り振られなければならない．

文字 $x\in\mathcal{X}$ の生起確率と符号長を $P(x)$ と $l(x)$ で表すと，その平均符号長 L は $L=\sum_{x\in\mathcal{X}}P(x)l(x)$ で与えられるが，ハフマン符号[2]は瞬時符号の中で最小の平均符号長を達成することができる．また，マクミランの定理[3]から，一意復号可能符号の符号長 $l(x)$ はクラフトの不等式 $\sum_{x\in\mathcal{X}}2^{-l(x)}\leq 1$ を満たさなければならず，クラフトの定理[4]から，符号長 $l(x)$ がクラフトの不等式を満たせば，その符号長を持つ瞬時符号を作ることができる．したがって，任意の一意復号可能符号に対して，それと同じ符号長を持つ瞬時符号を作れるため，瞬時符号の中で最適なハフマン符号は一意復号可能符号の中でも最適な符号となる．

その結果，1956 年にマクミランの定理が証明されて以来，約 60 年にわたりハフマン符号より圧縮率の良い一意復号可能な FV 符号を作る試みは全くされていなかった．しかし，マクミランの定理は一つの符号木を用いる一意復号可能符号に対して証明されているため，複数の符号木を用いることにより，ハフマン符号より平均

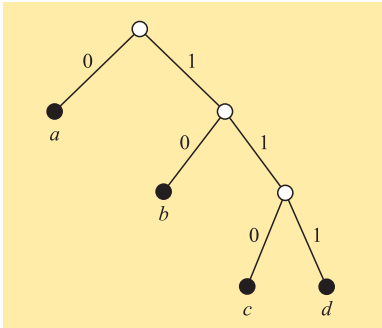


図 1　瞬時符号（語頭符号）の符号木の例　　文字 x の符号語は，根から x までのパスで与えられる．

山本博資　正員：フェロー　東京大学新領域創成科学研究科複雑理工学専攻
E-mail　hirosuke@ieee.org
Hirosuke YAMAMOTO, Fellow (School of Frontier Sciences, The University of Tokyo, Kashiwa-shi, 277-0871 JAPAN).
電子情報通信学会誌　Vol.104 No.1 pp.35-42 2021 年 1 月

符号長の小さい一意復号可能符号を構成できる可能性がある．実際，2015 年に Yamamoto, Tsuchihashi, Honda は，2 個の符号木を用いる AIFV 符号を提案し，ハフマン符号より小さい平均符号長を達成できることを示した[5]．瞬時符号ではない一意復号可能符号は，一般には大きな復号遅延が生じる可能性があり，そのような符号は実用的ではない．AIFV 符号では，復号遅延が 2 ビット以下になるように作られており，この特性から「準瞬時（almost instantaneous）」と名付けられている．

本稿では，AIFV 符号の基礎と拡張に関して紹介する．2. で AIFV 符号木及び符号化・復号アルゴリズムを説明し，3. で AIFV 符号の平均符号長がハフマン符号の平均符号長より短くなることを示す．次に，4. で最適な AIFV 符号木を構成するための反復最適化手法を紹介する．更に，5. で AIFV 符号木の種類の総数を，また 6. で m 個の符号木を使う AIFV-m 符号を取り扱う．最後に 7. で，AIFV 符号の様々な拡張符号について紹介する．

2. AIFV 符号の符号木と符号化・復号アルゴリズム

AIFV 符号は，二つの符号木 (T_0, T_1) で構成される．$\mathcal{X}=\{a, b, c, d\}$ に対する AIFV 符号木の例を図 2 に示す．各符号木は，完全内部節点（○）や葉（●）だけでなく，マスタ節点（■）とスレーブ節点（□）と呼ばれる不完全内部節点を含むことができる．文字 $x\in\mathcal{X}$ は，葉とマスタ節点に割り振られ，完全内部節点とスレーブ節点には割り振られない．マスタ節点の子は必ずスレーブ節点であり，マスタ節点は孫と ‘00’ の系列で結ばれている．一方，符号木 T_1 の根の ‘0’ の子[注1]はスレーブ節点であり，そのスレーブ節点は ‘1’ の子のみを持つ．したがって，T_1 では，根から ‘00’ で始まるパスは存在しないが，‘1’ 及び ‘01’ で始まるパスは必ず存在する．

データ系列 $x_1x_2x_3\cdots$ の符号化は次のように行われる[5]．

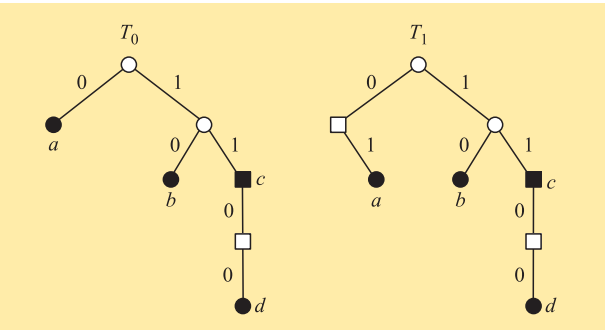


図 2　AIFV 符号の符号木の例　葉（●）とマスタ節点（■）に文字 $\{a, b, c, d\}$ が割り振られる．

［AIFV 符号の符号化アルゴリズム］

（1） 符号木 T_0 を用いて，最初の文字 x_1 を符号化する．

（2） x_i が葉（マスタ節点）で符号化された場合は，x_{i+1} は T_0（T_1）を用いて符号化する．

例えば，データ系列 *acdbaca* を図 2 の AIFV 符号を用いて符号化した場合，符号語系列は 0.11.1100.10.0.11.01 となる．なお「.」は符号語の区切りが分かりやすいように挿入してあるが，実際の符号語系列には，「.」は存在しない．このときの符号木の遷移は，$T_0\to T_0\to T_1\to T_0\to T_0\to T_0\to T_1$ となる．

与えられた符号語系列に対して，AIFV 符号の復号は次のように行われる[5]．

［AIFV 符号の復号アルゴリズム］

（1） 最初の文字 x_1 の復号には，符号木 T_0 を用いる．

（2） 現時点の符号木の根から，符号語系列に沿ってできるだけ深く，葉またはマスタ節点までたどる．たどり着いた葉またはマスタ節点に割り振られている文字 x_i を出力する．

（3） 符号語系列の語頭から，x_i の符号語を削除する．x_i が葉（マスタ節点）の場合は，x_{i+1} の復号には T_0（T_1）を用いる．

例えば，図 2 の AIFV 符号において，現時点の符号木が T_0 で，符号語系列が 110010… の場合は，d が復号される．その次の文字は，d の符号語 1100 が削除された符号語系列 10… に対して T_0 を用いて復号することになる．また，現時点の符号木が T_0 で符号語系列が 1101… の場合は，c が復号される．その次の文字は，c の符号語 11 が削除された符号語系列 01… に対して T_1 を用いて復号することになる．同様にして，符号語系列 01111001001101 から，データ系列 *acdbaca* が復号できる．

上記の例のように，T_0 において符号語系列が 1100… の場合は，d の符号語 1100 を読み終えた時点で d が瞬時に復号できる．しかし，符号語系列が 1101… の場合は，c の符号語 11 より 2 ビット長い 1101 を読んだ時点で c が復号される．そのため，AIFV 符号の復号では，最大 2 ビットの遅延が生じる．

3. AIFV 符号の平均符号長

次に，AIFV 符号の平均符号長 L_{AIFV} を評価する．符

（注 1）　親節点と ‘0’ で結ばれた子節点を，「親節点の ‘0’ の子」と表記する．

号木 T_j，$j=0,1$，の平均符号長と定常確率を L_j と Q_j で表すと，平均符号長 L_{AIFV} は次式で与えられる．

$$L_{\mathrm{AIFV}}=Q_0L_0+Q_1L_1 \qquad (1)$$

$$=\frac{Q_{1,0}}{Q_{0,1}+Q_{1,0}}L_0+\frac{Q_{0,1}}{Q_{0,1}+Q_{1,0}}L_1 \qquad (2)$$

ここで，$Q_{0,1}$ は T_0 から T_1 への遷移確率であり，$Q_{1,0}$ は T_1 から T_0 への遷移確率である．符号木 T_0 においてマスタ節点に割り当てられている文字の集合を $\mathcal{M}_0$，符号木 T_1 において葉に割り振られている文字の集合を $\mathcal{L}_1$ で表し，$x\in\mathcal{X}$ の生起確率を $P(x)$ で表すと，符号木の遷移確率は次式で与えられる．

$$Q_{0,1}=P(\mathcal{M}_0)=\sum_{x\in\mathcal{M}_0}P(x) \qquad (3)$$

$$Q_{1,0}=P(\mathcal{L}_1)=\sum_{x\in\mathcal{L}_1}P(x) \qquad (4)$$

例として，$\mathcal{X}=\{a,b,c,d\}$ で，$P(a)=0.45$，$P(b)=0.3$，$P(c)=0.2$，$P(d)=0.05$ の場合を考える．この確率分布のエントロピーは $H(P)=-\sum_{x\in\mathcal{X}}P(x)\log_2 P(x)\approx1.7200$ である．また，この分布に対するハフマン符号の符号木は図 1 で与えられ，その平均符号長は $L_{\mathrm{H}}=1.8$ である．これに対して，図 2 の AIFV 符号では，T_0 と T_1 の平均符号長は $L_0=1.65$ と $L_1=2.1$ である．T_0 では内部節点であるマスタ節点にも文字 $x\in\mathcal{X}$ が割り振られているため，一般に L_0 は L_{H} より小さくできる．一方，T_1 では 00 から始まるパスが存在しないため，L_1 は L_{H} より大きくなる．符号木の遷移確率は，式（3），（4）から，$Q_{0,1}=P(c)=0.2$，$Q_{1,0}=P(a)+P(b)=P(c)=0.8$ となり，定常確率は $Q_0=0.8$ と $Q_1=0.2$ となる．よって，式（1）から，AIFV 符号の平均符号長は $L_{\mathrm{AIFV}}=1.65\cdot0.8+2.1\cdot0.2=1.74$ で与えられる．ハフマン符号の平均符号長 $L_{\mathrm{H}}=1.8$ より小さい平均符号長を達成できていることが分かる．

他の例として，次に $\mathcal{X}=\{a,b,c\}$ で，$P(a)=0.9$，$P(b)=0.05$，$P(c)=0.05$，$H(P)\approx0.5690$ の場合を考える．この例のようにある一つの文字の生起確率が 1 に近いと，エントロピー $H(P)$ は 1 より小さくなるが，ハフマン符号の平均符号長は 1 より小さくなれない．これに対して，図 3 のように，AIFV 符号では T_0 の根をマスタ節点にすることができる．この AIFV 符号は，$L_0=0.3$，$L_1=1.2$，$Q_{0,1}=0.9$，$Q_{1,0}=1$，$Q_0=10/19$，$Q_1=9/19$ を持ち，平均符号長は $L_{\mathrm{AIFV}}\approx0.7263$ と 1 より小さくなる．

文字 a は T_0 の根に割り振られているため，その符号語は長さ 0 の空系列となる．空系列を λ で示すと，例えば，データ系列 $aaab$ の符号語系列は，$\lambda.1.\lambda.010=1010$ となる．復号は次のようになる．T_0 の根から，符号語系列 1010 に沿ってたどろうとしてもそのようなパスは存在しない．その結果，最初の文字が根に割り振られた a であることが分かる．a の根はマスタ節点であるため，次の文字は符号語系列 1010 に対して T_1 を用いて復号する．このように，T_0 に長さ 0 の符号語が存在しても，正しく復号が行える(注2)．

FV 符号の平均符号長 L とエントロピー $H(P)$ に対して，冗長度を $D=L-H(P)$ で定義する．また，全ての確率分布 P に対する冗長度の最悪値である最悪冗長度を $D^*=\sup_P D$ で定義する．このとき，任意の確率分布に対して，ハフマン符号の冗長度 D_{H} は $0\le D_{\mathrm{H}}<1$ を満たし，最悪冗長度 D_{H}^* は $P_{\max}=\max_{x\in\mathcal{X}}P(x)\to1$ である確率分布で生じ，$D_{\mathrm{H}}^*=1$ となる．これに対して，最適な AIFV 符号の冗長度 D_{AIFV} は，任意の確率分布に対して $0\le D_{\mathrm{AIFV}}<0.5$ を満たし，最悪冗長度 D_{AIFV}^* は $P_{\max}\to1$ である確率分布で生じ，$D_{\mathrm{AIFV}}^*=0.5$ となる(6)．

表 1 に，ハフマン符号と最適な AIFV 符号の比較を示す．ハフマン符号も，2 文字 $x_1x_2\in\mathcal{X}^2$ を一文字とみなして符号化すると，最悪冗長度を 0.5 に減らすことができる．しかし，$\mathcal{X}$ のサイズを $|\mathcal{X}|$ で表すと符号木サイズが $O(|\mathcal{X}|^2)$ になり，また，2 文字組みの最初の文字に対する符号化遅延と復号遅延が大きくなる．これに対して，AIFV 符号を用いれば，符号木サイズ $O(2|\mathcal{X}|)$ と最大復号遅延 2 ビットで最悪冗長度 0.5 を実現できる．

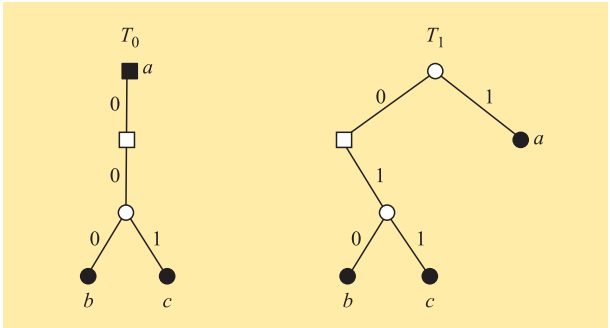


図 3　AIFV 符号の符号木の例　T_0 の根に a が割り振られている．

表 1　ハフマン符号と AIFV 符号の比較

	ハフマン符号	$\mathcal{X}^2$ に対するハフマン符号	AIFV 符号
最悪冗長度	1	0.5	0.5
符号木サイズ	$O(\lvert\mathcal{X}\rvert)$	$O(\lvert\mathcal{X}\rvert^2)$	$O(2\lvert\mathcal{X}\rvert)$
復号遅延	0	$x_1x_2\in\mathcal{X}^2$ の x_1 に対して大	最大 2 ビット

(注 2)　詳しくは，符号語系列の最後に空系列 λ が含まれているか否かを判断するための工夫が必要となる(5)．

4. 最適な AIFV 符号木の構成法

与えられた情報源の確率分布 $\{P(x), x\in\mathcal{X}\}$ に対して，瞬時符号の中で最小の平均符号長を持つハフマン符号木は，ハフマンアルゴリズム[2]により容易に求めることができる．これに対して，平均符号長 L_{AIFV} を最小にする最適な AIFV 符号木の構成は容易ではない．これは，葉に加えてマスタ節点にも文字 $x\in\mathcal{X}$ を割り振ることができる二つの符号木 (T_0, T_1) を同時に最適化しなければならないためである．しかし，下記の反復最適化手法を用いることにより最適な AIFV 符号木を求めることができる[5]．

与えられた符号木対 (T_0, T_1) に対して，C を次のように定義する．

$$C=\frac{L_1-L_0}{Q_{0,1}+Q_{1,0}} \tag{5}$$

$L_1\geq L_0$ から，$C\geq 0$ である．この C を用いると，式(2)で与えられる L_{AIFV} は次のように表現できる．

$$\begin{aligned} L_{\mathrm{AIFV}}&=L_0+CQ_{0,1}=L_0+CP(\mathcal{M}_0) &(6)\\ &=L_1-CQ_{1,0}=L_1-CP(\mathcal{L}_1) &(7)\end{aligned}$$

L_{AIFV} は T_0 と T_1 の両方に依存するが，C が固定されているときは，式(6)から符号木 T_0 のみを用いて表すことができ，同時に式(7)から符号木 T_1 のみを用いて表現できる．この特徴を利用して，下記の反復最適化手法により，平均符号長 L_{AIFV} を最小にする最適な符号木対 (T_0, T_1) を求めることができる．

[AIFV 符号木の反復最適化手法]

（1） [初期化] $C=C_{\mathrm{init}}$ に設定する．

（2） 与えられた C に対して，$L_0+CP(\mathcal{M}_0)$ を最小にする T_0 と，$L_1-CP(\mathcal{L}_1)$ を最小とする T_1 を求める．

（3） 得られた T_0 と T_1 に対して，L_0，L_1，$Q_{0,1}=P(\mathcal{M}_0)$，$Q_{1,0}=P(\mathcal{L}_1)$ を求め，式(5)により新しい C を求める．

（4） 新しい C の値がステップ(2)で用いた C の値と同じ場合は終了．C が変化した場合は，新しい C を持ってステップ(2)に戻る．

C の初期値 C_{init} はどのような正の値でも構わないが，T_0 及び T_1 の深さ 2 の節点を根とする部分木が一様な構造を持つ理想的な場合の C の値である $C_{\mathrm{init}}=2-\log_2 3$ を用いると，収束が速い．

ステップ(4)で，C の値が変化するときは L_{AIFV} は 1 ラウンド前の値より必ず減少し，C の値が変化しないときは L_{AIFV} は最小値を達成している[5]．L_{AIFV} の厳密な単調減少性と，異なる符号木対 (T_0, T_1) が有限個しか存在しないことから，上記の反復最適化手法は必ず収束する．なお，ステップ(2)における $L_0+CP(\mathcal{M}_0)$ 及び $L_1-CP(\mathcal{L}_1)$ の最小化には，整数計画法[5]や動的計画法[7,8]などを用いることができる．

5. AIFV 符号の符号木数

アルファベットサイズ $n=|\mathcal{X}|$ に対して，全二分木（full binary tree）を使うハフマン符号木の総数は，$n-1$ 次のカタラン数（Catalan number）C_{n-1} で与えられる[9]．ここで，n 次のカタラン数は次式で定義される．

$$C_n=\frac{1}{n+1}\binom{2n}{n}=2^{2n+o(n)} \tag{8}$$

これに対して，AIFV 符号木の符号木 T_0 の総数は，$n-1$ 次のシュレーダー数（Schröder number）(注3) S_{n-1} で与えられる[10]．ここで，S_n は次式で定義される．

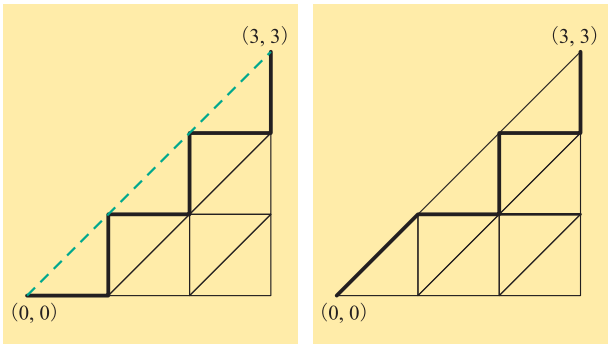
$$\begin{aligned} S_n&=\sum_{k=0}^{n}\frac{1}{n-k+1}\binom{2n-2k}{n-k}\binom{2n-k}{k}\\ &=(3+2\sqrt{2})^{n+o(n)}\approx 2^{2.5431n+o(n)} &(9)\end{aligned}$$

また，符号木 T_1 の総数は，$S_{n-1}-S_{n-2}$ で与えられる[10]．式(8)，式(9)から，C_n に比べて S_n の方が指数的に大きく，与えられた情報源の確率分布 $\{P(x)\}$ により良くマッチした符号木を選べるため，平均符号長をより小さくできる．

全二分木はダイクパス（Dyck path）と全単射の関係があることがよく知られているが[9]，T_0 はシュレーダーパスと全単射の関係がある[10]．図 4 に図 1 のハフマン符号木に対応したダイクパスと，図 2 の T_0 に対応したシュレーダーパスを示す．ダイクパスでは，$(n-1, n-1)$ から $(0, 0)$ へのパスにおいて，符号木の根から深さ優先探索で節点が左の子を持つときはダイクパスは下へ移動し，節点が子を持たないときはダイクパスは左に移動する．シュレーダーパスはダイクパスに追加して斜めの移動が可能であり，この斜めの移動は T_0 の節点がマスタ節点とスレーブ節点の対である場合に対応している[10]．

データ系列の確率分布が未知の場合は，データ系列の頻度分布を調べ，その頻度分布に適した AIFV 符号木を使う必要がある．その場合は，使用した AIFV 符号木の情報を符号語系列の前に付加しなければならない

(注3) 小シュレーダー数と区別する場合は，大シュレーダー数と呼ぶ．



(a) ダイクパス　　(b) シュレーダーパス

図 4　ダイクパスとシュレーダーパス　それぞれ図 1 の符号木と図 2 の符号木 T_0 に対応している．

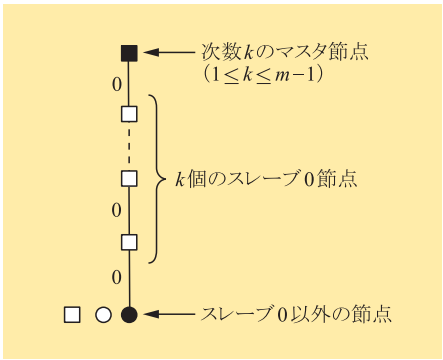
が，AIFV 符号木とシュレーダーパスの対応を利用して，AIFV 符号木の情報を二元符号語に符号化できる[10]．

上では，節点の左右の子を根とする部分木が異なる形状の場合，その左右の部分木を入れ替えた木を異なる木として取り扱っている．しかし，左右の部分木を入れ替えても符号長は変わらないため，圧縮性能は同じである．符号木の各節点で左右の部分木を入れ替えて得られる全ての木を同一とみなす場合をコンパクト木（compact tree）というが，コンパクト AIFV 符号木を数え上げる手法と，その手法を用いてコンパクト AIFV 符号木の情報を二元符号語に符号化する方法が知られている[11]．

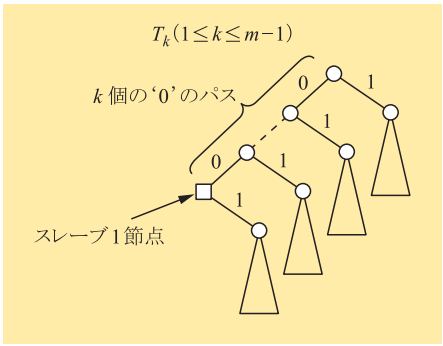
6. AIFV-m 符号

AIFV 符号は二つの符号木を用いて最大 2 ビットの復号遅延を許す FV 符号であるが，その拡張として，m 個の符号木 $(T_0, T_1, \cdots, T_{m-1})$ を使用し最大 m ビットの復号遅延を許す AIFV-m 符号を定義することができる[6]．$m=2$ の場合は，2. の AIFV 符号に一致する．

AIFV-m 符号では，$m-1$ 種類のマスタ節点を使用する．スレーブ節点のうち，‘0’ の子を持つものをスレーブ 0 節点，‘1’ の子を持つものをスレーブ 1 節点と呼ぶことにする．このとき，図 5(a) のようにマスタ節点の下に k 個のスレーブ 0 節点が続くマスタ節点を，次数 k のマスタ節点という(注4)．また，葉を簡単のため次数 0 のマスタ節点と呼ぶ．AIFV-m 符号の符号木は，完全内部節点と次数 0～$m-1$ のマスタ節点を含むことができる．‘0’ が k 個続くパスを ‘0^k’ で表すと，図 5(b) に示すように，符号木 T_k，$1 \le k \le m-1$ の根は，‘0^k’ のところにスレーブ 1 節点を持たなくてはならない．



(a) 次数 k のマスタ節点



(b) 符号木 T_k

図 5　次数 k のマスタ節点と符号木 T_k　次数 k のマスタ節点は ‘0^{k+1}’ のパスを持つが，T_k の根は ‘0^{k+1}’ のパスを持たない．

なお，符号木 T_0 の根は，次数 1～$m-1$ のマスタ節点になることができ，符号木 T_k，$2 \le k \le m-1$ の根は，次数 1～$k-1$ のマスタ節点になることができる．しかし，T_1 の根は，マスタ節点になることはできない．文字 $x \in \mathcal{X}$ は，マスタ節点に割り振られるが，完全内部節点とスレーブ節点には割り振られない．

データ系列 $x_1x_2\cdots$ に対して，符号化及び復号における符号木の遷移則は次のようになる．

［AIFV-m 符号木の遷移則］

（1） 最初の文字 x_1 の符号化（復号）には，符号木 T_0 を用いる．

（2） 文字 x_i の符号化（復号）が，次数 k のマスタ節点で行われた場合，x_{i+1} の符号化（復号）には，符号木 T_k を用いる．

例として，$\mathcal{X}=\{a, b, c, d\}$，$m=3$ の場合の符号木の例を，図 6 に示す．この符号木を用いて，データ系列 *accbd* を符号化すると，符号語系列は 0.11.11.01.1100＝01111011100 となる．

3. で述べたように，AIFV 符号の最悪冗長度は 0.5 であるが，AIFV-m 符号を用いると最悪冗長度を更に小さくできる．これは，$P_{\max}$ が大きい場合に，T_1 以外の符号木の根をマスタ節点にすることで，$P_{\max}$ を持つ文字の多くを，符号長 0 の空系列に符号化できるためであ

(注 4)　グラフ理論での節点の次数とは定義が異なる．

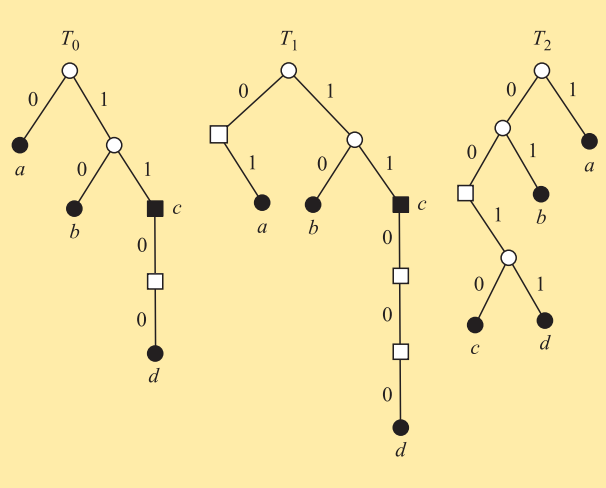


図 6　AIFV-m 符号の例　データ系列 *accbd* を符号化すると，符号語系列は 01111011100 となる．

表 2　AIFV-m 符号の平均符号長[13]（L_H：ハフマン符号，L_{A-m}：AIFV-m 符号）

	L_H	L_{A-2}	L_{A-3}	L_{A-4}	L_{A-5}	エントロピー
$P^{(1)}$	1.1500	0.7515	0.6551	0.6356	0.6345	0.5761
$P^{(2)}$	2.5630	2.5035	2.5003	2.5002	2.5000	2.4894

る．$2 \le m \le 5$ の場合の AIFV-m 符号の最悪冗長度は $1/m$ であり[6),12)]，また，任意の $m \ge 2$ に対して $P_{\max} \to 1$ のとき，AIFV-m 符号の冗長度が $1/m$ となる[6]ことが証明されている．例として，確率分布 $P^{(1)} = \{0.900, 0.050, 0.049, 0.001\}$ と $P^{(2)} = \{0.33, 0.3, 0.119, 0.1, 0.05, 0.04, 0.03, 0.02, 0.01, 0.001\}$ に対する AIFV-m 符号の平均符号長を表 2 に示す[13]．m が大きくなるに連れて，平均符号長を小さくできる．

与えられた確率分布 $\{P(x)\}$ に対して平均符号長を最小にする最適な AIFV-m 符号の符号木は，4. で説明した反復最適化手法の拡張アルゴリズムにより求めることができる[13]．

7. AIFV 符号の様々な拡張

本章の各節では，前章までに紹介した AIFV 符号や AIFV-m 符号に関連した様々な拡張符号やアルゴリズムについて紹介する．

7.1　動的 AIFV 符号

情報源の確率分布 $\{P(x)\}$ が未知の場合，データ系列を一文字読むごとにその文字の符号化とその時点までの頻度分布に基づく符号木の更新を行う動的符号が用いられる．そのような符号として，瞬時符号では動的ハフマン符号が知られている[14),15)]．

AIFV 符号の場合に対しても動的な符号化を考えることができるが，4. で述べたように，符号木が固定された静的な符号木の場合でも最適な符号木を求めるのは容易ではない．そのため，頻度分布に基づいて少ない計算量で符号木を最適に更新する符号化法を作るのは困難である．そこで，最適性は考慮しないが，頻度分布 $\{P(x)\}$ に対して冗長度が $D_{AIFV} = L_{AIFV} - H(P) < 0.5$ を満たすことを保証する次のような動的 AIFV 符号化アルゴリズムが知られている[16]．

［動的 AIFV 符号化アルゴリズム］

（1）［初期化］データ系列 $x_1x_2\cdots$ に対して，T_H を動的ハフマン符号の初期符号木とし，$i=1$ とする．
（2）T_H から，$D_{AIFV} < 0.5$ を満たすように AIFV 符号木 (T_0, T_1) を作る．
（3）(T_0, T_1) を用いて，x_i を AIFV 符号化する．
（4）x_i の頻度を一つ増やし，動的ハフマン符号の更新アルゴリズム[15]を用いて T_H を更新する．
（5）データ系列に文字が残っていれば，$i = i+1$ として，ステップ（2）に戻る．

なお，ステップ（2）の AIFV 符号木 (T_0, T_1) の作り方は，原論文[16]を参照して欲しい．また，$P_{\max}$ が 1 に近い場合は，AIFV-m 符号を用いて同様の動的符号化を行うことができる[16]．

7.2　アルファベティック AIFV 符号

任意のデータ系列に対して，データ系列のアルファベット順と符号語系列のアルファベット順が一致する符号を，アルファベティック符号（alphabetic code）という．アルファベティック符号を用いると，データ系列に対してアルファベット順でソートや検索などを行いたいときに，復号せずに符号語系列のままで行うことができる．

瞬時符号の中で，最小の平均符号長を持つアルファベティック符号は，Hu-Tucker のアルゴリズム[17]や Garsia-Wachs アルゴリズム[18]を用いて構成することができる．これに対して，アルファベティック AIFV 符号は，三つの符号木を用いて構成できる[19]．また，三つの最適なアルファベティック AIFV 符号木は，4. で説明した反復最適化手法を拡張することにより求められる[20]．更に，瞬時符号のアルファベティック符号木を利用して，最適ではないが，最適なアルファベティック瞬時符号より小さい平均符号長を持つアルファベティック AIFV 符号を簡易に生成することもできる[19]．

7.3　AIVF 符号

FV 符号は，データ系列を 1 文字（あるいは固定長）で区切り，可変長の符号語に符号化する符号であるが，VF 符号（Variable-to-Fixed length code）では，逆に

データ系列を可変長で区切り，固定長の符号語に符号化する．また，FV 符号は符号木で表されるのに対し，VF 符号は分節木（parsing tree）を用いて表すことができる．符号語数を M，データ系列の平均分節長を L_{VF} としたとき，VF 符号の符号化レート R_{VF} は，$R_{\mathrm{VF}}=\lceil \log_2 M \rceil / L_{\mathrm{VF}}$ で与えられる．そのため，VF 符号では平均分節長 L_{VF} が大きいほど圧縮率が良くなる．一つの分節木を用いる場合，与えられた符号語数 M に対して，最大の平均分節長 L_{VF} を持つ最適な符号として，タンストール符号[21]が知られている．

これに対して，Yamamoto と Yokoo[22]は，$|\mathcal{X}|-1$ 個の分節木を用いる準瞬時 VF 符号（AIVF 符号：Almost Instantaneous VF code）を定義し，AIVF 符号がタンストール符号より大きい平均分節長 L_{VF} を達成できることを示した．

Yamamoto-Yokoo の論文[22]では，分節木の根に符号語を割り振ることは考えられていなかったが，根にも符号語を割り振ることを許し，動的計画法を用いることで，最大の平均分節長を持つ分節木を個別に求めることができる[23]．更に，その動的計画法に反復最適化手法を組み合わせることで，全体としてより大きい平均分節長を持つ AIVF 符号を構成できる[24]．

AIVF 符号では，$|\mathcal{X}|-1$ 個の分節木が用いられるが，これらの分節木を一つの仮想多重 AIVF 木（virtual multiple AIVF tree）に多重化することでメモリ使用量を削減することができる[25),(26]．

7.4 AIVF 符号に基づくユニバーサル符号

AIVF 符号では，情報源の確率分布 $\{P(x)\}$ が既知であるとして分節木が構成される．これに対して，確率分布が未知であるデータ系列を，分節木を適応的に成長させながら圧縮する様々なユニバーサル符号[27]が提案されているが，AIVF 符号で用いられている符号化手法に基づいたユニバーサル符号を作ることができる[28]．

AIVF 符号では複数の分節木を用いることで平均分節長が長くなるように工夫されているが，このユニバーサル符号では，仮想多重 AIVF 木とは異なる手法で，その工夫を一つの分節木上で実現している．また，このユニバーサル符号は，よく知られている LZW 符号（Welch 符号）[29]の改良符号とみなすこともでき，LZW 符号よりも良い圧縮率を達成できる[28]．

7.5 反復最適化手法の一般化

4. で述べた最適な AIFV 符号木を求める反復最適化手法を拡張することにより，AIFV-m 符号木の最適化[13]，アルファベティック AIFV 符号木の最適化[20]，AIVF 分節木の最適化[24]を行うことができるが，更に，一般の有限状態マルコフ連鎖の平均性能の最適化に用いることができる[30),(31]．

m 個の状態集合を $\mathcal{S}=\{s_0, s_1, \cdots, s_{m-1}\}$ とし，各状態 s_i は有限離散パラメータ t_i を持っているものとする．状態 s_i における性能は $f_i(t_i)$ で与えられるものとし，状態 s_i から状態 s_j への遷移確率 $Q_{i,j}(t_i)$ も t_i のみに依存する場合を考える．$\boldsymbol{t}=(t_0, t_1, \cdots, t_{m-1})$ のときの状態 s_i の定常確率を $Q_i(\boldsymbol{t})$ とすると，マルコフ連鎖全体の平均性能 $F(\boldsymbol{t})$ は，

$$F(\boldsymbol{t})=\sum_{i=0}^{m-1} Q_i(\boldsymbol{t}) f_i(t_i) \tag{10}$$

で与えられる．このシステムに対する平均性能の最適化問題 $\min_{\boldsymbol{t}} F(\boldsymbol{t})$（または $\max_{\boldsymbol{t}} F(\boldsymbol{t})$）を考える．

AIFV-m 符号の場合は，t_i が符号木 T_i に対応し，$f_i(t_i)$ が T_i の平均符号長 L_i に対応する．

このとき，式（5）の C と同様に，補助変数 $C_{i,i+1}$，$0\le i\le m-2$ を導入すると，$F(\boldsymbol{t})$ を各状態 s_i のパラメータ t_i だけを用いて表現できる．そして，$\{C_{i,i+1}\}$ が与えられている条件の下で，その表現を用いて s_i ごとに最適な t_i を求め，次に得られた $\boldsymbol{t}$ を用いて $C_{i,i+1}$ を更新するという反復最適化を行うことにより，$F(\boldsymbol{t})$ の最適化を実行することができる[31]．

7.6 その他の関連する話題

4. 及び 7.5 で述べた反復最適化手法において，反復回数は有限回数で終了することが示されている．これに対して，情報源の確率が有限桁であることを仮定して，二分探索を用いることで多項式時間の反復回数で終了することが証明できる[32]．なお，二分探索を用いなくても，反復最適化手法[5),(13),(20]の実装における反復回数は少ない．また，AIFV 符号の最適化を A^* アルゴリズムで解くこともできる[33]．

AIFV 符号及び AIFV-m 符号は，情報源の任意の定常無記憶な確率分布 $\{P(x)\}$ を対象にしているが，あらかじめ $\{P(x)\}$ のクラスが限定されているときは，符号木の形状を事前に決めておくことができる．例えば，$\{P(x)\}$ が幾何分布の場合には，AIFV-m と異なる複数の符号木を用いて Golomb-Rice 符号の拡張符号を構成できる[34]．

二元 AIFV 符号のクラスは，演算精度が 2 ビットの二元算術符号のクラスと対応付けることができる[33]．また，本稿では符号語に $\{0, 1\}$ を用いる二元 AIFV 符号について述べたが，多元 AIFV 符号も同様に構成することができる[5]．

8. ま と め

本稿では，AIFV 符号及びその拡張符号について説明し，ハフマン符号より小さい圧縮率を達成できることを

紹介した．ハフマン符号は単独で用いられることは少ないが，様々な圧縮ツールやユニバーサルデータ圧縮符号の中などで利用されている．そのような場合，ハフマン符号を AIFV 符号で置き換えることで圧縮性能を改善できる．特に，エントロピー$H(P)$が 1 より小さい場合，$\mathcal{X}^m$ に対してハフマン符号を用いている場合，復号遅延を高々 m ビットに限定したい場合などでは，圧縮率，復号遅延，メモリ量の削減の観点から，AIFV-m 符号の使用が勧められる．

謝辞　本稿に対して有益なコメントを頂いた福井大学岩田賢一准教授に感謝する．本稿で紹介した研究成果の一部は，JSPS 科研費 18H01436，20K11674 の助成を受けて実施したものである．

文　　献

（1） 山本博資，古賀正樹，有村光晴，岩本　貢，情報理論 基礎と広がり，共立出版，July 2012.
（2） D.A. Huffman, “A method for the construction of minimum-redundancy codes,” Proc. IRE, vol. 40, no. 9, pp. 1098-1102, Sept. 1952.
（3） B. McMillan, “Two inequalities implied by unique decipherability,” IRE Trans. Inf. Theory, vol. IT-2, no. 4, pp. 115-116, Dec. 1956.
（4） L.G. Kraft, “A device for quantizing, grouping, and coding amplitude-modulated pulses,” Master’s thesis, Dept. of Electrical Eng., MIT, 1949.
（5） H. Yamamoto, M. Tsuchihashi, and J. Honda, “Almost instantaneous fixed-to-variable length codes,” IEEE Trans Inf. Theory, vol. 61, no. 12, pp. 6432-6443, Dec. 2015.
（6） W. Hu, H. Yamamoto, and J. Honda, “Worst-case redundancy of optimal binary AIFV codes and their extended codes,” IEEE Trans Inf. Theory, vol. 63, no. 8, pp. 5074-5086, Aug. 2017.
（7） K. Iwata and H. Yamamoto, “A dynamic programming algorithm to construct optimal code trees of AIFV codes,” 2016 International Symposium on Information Theory and Its Applications（ISITA2016）, pp. 672-676, Monterey, U.S.A., Oct. 2016.
（8） M. Golin and E. Harb, “Speeding up the AIFV-2 dynamic programs by two orders of magnitude using range minimum queries,” arXiv : 2002.09885, Feb. 2020.
（9） R.P. Stanley, Catalan Numbers, Cambridge University Press, New York, 2015.
（10） K. Sumigawa and H. Yamamoto, “Coding of binary AIFV code trees,” 2017 IEEE Int. Symp. on Inf. Theory（ISIT2017）, pp. 1152-1156, Aachen, Germany, June 2017.
（11） K. Hashimoto, K. Iwata, and H. Yamamoto, “Enumeration and coding of compact code trees for binary AIFV codes,” 2019 IEEE Int. Symp. on Inf. Theory（ISIT2019）, pp. 1527-1531, Paris, France, July 2019.
（12） R. Fujita, K. Iwata, and H. Yamamoto, “On a redundancy of AIFV-m codes for m＝3, 5,” 2020 IEEE Int. Symp. on Inf. Theory（ISIT2020）, pp. 2373-2377, Los Angeles, U.S.A., June 2020.
（13） K. Iwata and H. Yamamoto, “An iterative algorithm to construct optimal binary AIFV-m codes,” 2017 IEEE Inf. Theory Workshop（ITW2017）, pp. 519-523, Kaohsiung, Taiwan, Nov. 2017.
（14） D. Knuth, “Dynamic Huffman coding,” J. Algorithms, vol. 6, no. 2, pp. 163-180, June 1985.
（15） J.S. Vitter, “Design and analysis of dynamic Huffman codes,” J. ACM, vol. 34, no. 4, pp. 825-845, Oct. 1987.
（16） T. Hiraoka and H. Yamamoto, “Dynamic AIFV coding,” 2018 International Symposium on Information Theory and Its Applications（ISITA2018）, pp. 577-581, Singapore, pp. 28-31, Oct. 2018.
（17） T.C. Hu and A.C. Tucker, “Optimal computer search trees and variable length alphabetic codes,” SIAM J. Appl. Math., vol. 21, no. 4, pp. 514-532, Dec. 1971.
（18） A.M. Garsia and M.L. Wachs, “A new algorithm for minimum cost binary trees.” SIAM J. Comput., vol. 6, no. 4, pp. 622-642, 1977.
（19） T. Hiraoka and H. Yamamoto, “Alphabetic AIFV codes constructed from hu-tucker codes,” 2018 IEEE Int. Symp. on Inf. Theory（ISIT2018）, pp. 2182-2186, Vail, U.S.A., June 2018.
（20） 中西雄也，藤田龍星，岩田賢一，山本博資，“2 元アルファベッティク AIFV 符号における最適な符号木の構成法，” 信学技報，IT2018-62, SIP2018-92, RCS2018-269, pp. 155-160, Jan. 2019.
（21） B. Tunstall, Synthesis of noiseless compression codes, Ph. D. thesis, Georgia Institute of Technology, 1967.
（22） H. Yamamoto and H. Yokoo, “Average-sense optimality and competitive optimality for almost instantaneous VF codes,” IEEE Trans Inf. Theory, vol. 47, no. 6, pp. 2174-2184, Sept. 2001.
（23） D. Dubé and F. Haddad, “Individually optimal single- and multiple-tree almost instantaneous variable-to-fixed length codes,” 2018 IEEE Int. Symp. on Inf. Theory（ISIT2018）, pp. 2192-2196, Vail, U.S. A, June 2018.
（24） 牧野楓也，藤田龍星，岩田賢一，D. Dubé，山本博資，“平均性能最適化の繰り返しアルゴリズムによる AIVF 符号の改良，” 第 42 回情報理論とその応用シンポジウム（SITA2019）, pp. 277-282, Nov. 2019.
（25） S. Yoshida and T. Kida, “An efficient algorithm for almost instantaneous VF code using multiplexed parse tree,” Data Compression Conf. 2010（DCC 2010）, pp. 219-228, Snowbird, U.S.A., March 2010.
（26） S. Yoshida and T. Kida, “An efficient variable-to-fixed length encoding using multiplexed parse trees,” J. Discret. Algorithms, vol. 32, pp. 75-86, May 2015.
（27） D. Salomon and G. Motta, Handbook of Data Compression, Fifth Edition, Springer-Verlag, London, 2010.
（28） H. Yamamoto, K. Imaeda, K. Hashimoto, and K. Iwata, “A universal data compression scheme based on the AIVF coding techniques,” 2020 IEEE Int. Symp. on Inf. Theory（ISIT2020）, pp. 2378-2382, Los Angeles, U.S.A., June 2020.
（29） T.A. Welch, “A technique for high-performance data compression,” IEEE Computer, vol. 17, no. 6, pp. 8-19, June 1984.
（30） 山本博資，“AIFV 符号の反復構成法に対する最適性証明，” 信学技報，IT2014-57, pp. 19-22, Jan. 2015.
（31） R. Fujita, K. Iwata, and H. Yamamoto, “An iterative algorithm to optimize the average performance of Markov chains with finite states,” 2019 IEEE Int. Symp. on Inf. Theory（ISIT2019）, pp. 1902-1906, Paris, France, July 2019.
（32） M. Golin and E. Harb, “Polynomial time algorithms for constructing optimal binary AIFV-2 codes,” arXive : 2001.11170, Jan. 2020.
（33） 内田尚大，西新幹彦，“A* アルゴリズムを用いた順序保存性のない最適な算術符号の探索について，” 信学技報，IT2018-30, pp. 19-23, Sept. 2018.
（34） R. Sugiura, Y. Kamamoto, N. Harada, and T. Moriya, “Optimal Golomb-Rice code extension for lossless coding of low-entropy exponentially distributed sources,” IEEE Trans Inf. Theory, vol. 64, no. 4, pp. 3153-3161, Jan. 2018.

（2020 年 7 月 13 日受付　2020 年 7 月 22 日最終受付）

山本　博資（やまもと　ひろすけ）（正員：フェロー）
昭 50 静岡大・工・電気卒．昭 55 東大大学院博士課程了．工博．徳島大・工，電通大，東大・工／情報理工／新領域を経て，現在，東大名誉教授，早大・基幹理工・客員教授，中大客員機構教授，明大客員研究員．データ圧縮アルゴリズムをはじめとする情報理論全般の研究に従事．平 21 年度本会論文賞，平 29 年度本会業績賞各受賞．IEEE 会員（Life-Fellow）．