利用目的による系列の周期探索アルゴリズム選択

宮崎 武[1,a)]　荒木 俊輔[2]　上原 聡[1]

概要：系列の周期は，繰り返し写像による生成系列の重要な性質の一つである．擬似乱数生成器などを構成する場合，長周期であることが求められるが，このような系列長が長い系列においては，その周期を求めることは容易ではない．本稿では，一般的に良く利用されている周期探索アルゴリズムである Floyd 法と Brent 法について調査した．これらのアルゴリズムは，探索途中の写像値を必要最低限のみ記憶するという特徴を持っている．そのため，必要なメモリ量は小さいが，同一の入力値に対する写像演算を複数回繰り返す必要がある．一方，現在ではメモリ容量が向上したことから，探索途中の写像値を全て記憶することも可能となってきている．そこで，我々は Floyd 法と Brent 法を下敷きとして，写像値を全て記憶する方式について検討し，必要となるメモリ量について理論的な考察を行う．また，これら複数のアルゴリズムについて，どのような場面でどのアルゴリズムを採用すべきか，という利用用途に合わせた最適なアルゴリズム選択について考察する．

キーワード：周期探索アルゴリズム，Floyd 法，Brent 法

A Selection of Cycle-Finding Algorithms for Sequences with Their Purposes of Use

Takeru Miyazaki[1,a)]　Shunsuke Araki[2]　Satoshi Uehara[1]

Abstract: Period of sequences is one of the most important properties for iterative sequences. It is not easy to find the period of a sequence in which period is unknown. In this present paper, we investigate Floyd's and Brent's method as two famous cycle-finding ones. They have a characteristic property of recording minimum elements which are mapped values in the finding methods. Though the number of memories required by the methods is small, they must calculate a plural number of mapping in which an input value is the same. Whereas, since nowadays computers have a huge amount of memories, we can record all mapped values in these methods. Then, we consider new methods recording all mapped values, which is based on Floyd's and Brent's method, and analyze theoretical values on required memory. Moreover, we also consider an optimal method selecting with its purpose of use.

Keywords: Cycle-Finding algorithm, Floyd's method, Brent's method

1. はじめに

周期は，系列を用いたアプリケーションの設計において重要な指標の一つである．有限要素の出力値が次の入力値となる繰り返し写像による生成系列において，系列の途中からはある部分系列の繰り返しによって構成される．このような写像を用いた擬似乱数生成器の設計では，周期は重要な検討事項となる．

m 系列のように，理論的に周期が証明されているものについてはこのような周期の探索は不要である．このような一部の例外を除くと，一般的に系列の周期は周期探索アルゴリズムによって調査しなければわからない．整数上のロジスティック写像 [3], [4] のように繰り返し系列の理論的な

[1] 北九州市立大学, 福岡県北九州市若松区ひびきの 1–1,
1–1 Hibikino, Wakamatsu, Kitakyushu City, Fukuoka 808–0135, Japan
[2] 九州工業大学, 福岡県飯塚市川津 680–4,
680–4 Kawazu, Iizuka City, Fukuoka 820–8502, Japan
[a)] miyazaki@kitakyu-u.ac.jp

解析が難しい系列では，必ずしも事前に周期が判るわけではないので，このような周期は実際に写像演算を繰り返して系列を生成してみなければ不明であることが多い．

また，系列の周期を探索する場合は，その利用目的に合わせた様々な状況が考えられる．例えば，長周期であることが予測できるある特定の系列において，その周期を確認する場合にはより高速なアルゴリズムが求められる．一方で，周期の平均値を求めるような場合では，多くの初期条件を変化させながらそれぞれの生成系列について各々周期を求める必要があるため，異なるパラメータに対して独立した調査を行うのに適したアルゴリズムを選択するべきである．

良く知られている周期探索アルゴリズムとしては，Floyd 法 [1] と Brent 法 [2] が挙げられる．Floyd 法は 2 つの要素のみを記憶しながら系列の周期とリンク長を求めるアルゴリズムである．記憶量が少なく，そのため比較回数も少ないが同じ写像の計算を繰り返す必要があり効率的ではないと言われている．一方，Brent 法は Floyd 法と比較して記憶しておく要素は増えるが，無駄な写像計算を減らして効率的に周期とリンク長を計算できる．

本稿では，これらの系列探索アルゴリズムについて周期探索に必要な写像回数の理論的な解析を行う．その結果を用いて，必要なメモリ量を算出し，有限要素の系列における写像値記録版の Floyd 法，Brent 法の実装を行う．また，Floyd 法よりも高速な Brent 法については，さらに通常版と写像値記録版を組み合わせた併用版 Brent 法や，内部の更新間隔を 2 倍毎から 3 倍毎に変更した 3 倍版 Brent 法を提案する．これらの提案方式について，幾つかの繰り返し写像による系列を用いて数値実験を行い探索速度を比較する．これらの結果を踏まえ，周期探索目的に合わせた周期探索アルゴリズムの選択法について考察する．

2. 周期と探索アルゴリズム

ここでは，本稿における系列の周期・リンク長の定義と既知の周期探索アルゴリズムについて説明する．

2.1 系列と周期，リンク長

写像 $M()$ の生成系列 $S = (x_0, x_1, x_2, \cdots)$ を，漸化式 $x_{i+1} = M(x_i),\ i = 0, 1, 2\cdots$ で定義する．ここで，系列の先頭に当たる要素 x_0 を初期値といい IV で表現する．

このような系列において，$x_i = x_j, i < j$ を満たす要素があればこの系列は途中からこの系列の部分系列 $(x_i, x_{i+1}, \cdots, x_{j-1})$ を繰り返し出力する．本稿では，この繰り返し出力される部分系列のうち長さが最小となるものを**ループ**と呼び，ループ中に含まれる要素数を**周期**と呼ぶ．また，初期値 x_0 がループ内に含まれる要素ではない場合，この系列は部分系列の繰り返し部分に到達する前に非周期な部分系列が存在する．この非周期部分系列を**リンク**と呼び，リンクに含まれる要素数を**リンク長**と呼ぶ．ここで，もしリンクが存在しない場合，すなわち初期値 x_0 がループ内に存在する場合，リンク自身は存在していないが便宜上リンク長を 0 と定義する．また，系列がある不動点に到達した場合，つまり $x_k = M(x_k)$ のように入力値と写像値が同一となった場合もこの部分を周期 1 のループとして取り扱う．

2.2 周期探索

一般に系列 S の周期 P を求めることは困難である．系列の中には m 系列や Knuth の二次系列 [1]，ある一定の条件を満たす素体上のロジスティック写像 [5], [6] など，理論的な解析を行うことができて周期が事前に判明しているものもあるが，それらの系列を除くと実際に写像演算を行って系列を生成しない限り系列の周期を求めることは難しい．

また実際に系列の要素 x_i を計算しながらその周期 P やリンク長 L を求める場合，最も単純な方法としては以下のような手順が考えられる．

(1) $i = 1$, 部分系列 $S' = (x_0), X_0$ は初期値とする
(2) $x_i = M(x_{i-1})$ を S' の末尾に加える
(3) $j = 0$ から $i - 1$ まで以下を繰り返す
　もし $x_j = x_i$ ならば，周期 $P = (i - j)$, リンク長 $L = j$ として終了する
(4) i を 1 つ増やして (2) に戻る

容易に想像できるように，このアルゴリズムは非常に多数の記憶領域と比較回数が必要となり，あまり効率的だとは言えない．よって，擬似乱数生成器に用いる系列のような長い周期・リンク長が見込まれる系列での周期探索には，より効率的なアルゴリズムを用いる必要がある．

2.3 Floyd 法

Floyd 法は，R. Floyd によって有向グラフの周期を発見する効率的なアルゴリズムとして提案され，Knuth の著書によって系列の周期探索アルゴリズムとして紹介されている [1]．また，このアルゴリズム内に出現する 2 要素を亀と兎に例えて，兎亀算や Floyd's Tortoise and Hare と呼ばれることもある．

Floyd 法の特徴は，記憶しておく要素数が 2 個のみと少ないことが挙げられる．現状の PC においては十分なメモリ量が確保できるため，このような要素数が少ない事は直接有効であるという場面は限られる．しかし，記憶しておく要素が低いことが比較回数が少ないことを意味し，また楕円曲線上の点のような 1 つの要素を表現するのに複数の値を記憶しておく必要があるものなどこのような特徴の恩恵を受けるものも少なからず存在している．

一方で，Floyd 法の欠点としては記憶しておく要素は少ない分同じ写像演算を繰り返し行う必要があることと，アルゴリズムの第 1 段階では周期 P の値が直接導出されず，

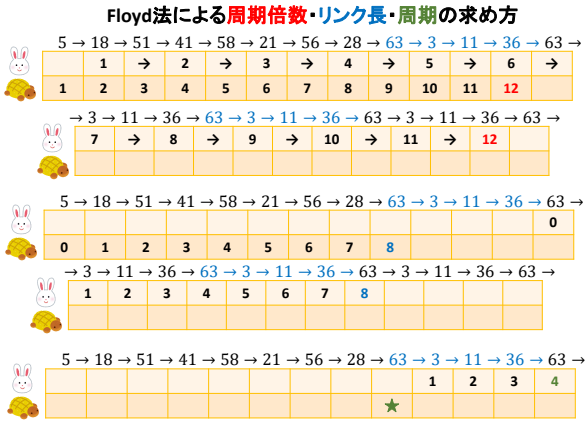

図 1　Floyd 法による周期リンク長探索例

P の整数倍である周期倍数 $P' = mP$ (ただし m は整数) がまず求まることである．この P' は $m = 1$ の場合，つまり P そのものである場合も考えられるが，一般的にどのような条件で P' が P と等しくなるかは不明で，アルゴリズムを最後まで行わなければ P を導出することができない．

以下に Floyd 法のアルゴリズムを示す．記憶しておく要素を T(Tortoise) と H(Hare) とする．また初期値 $IV = x_0$ とする．

(1) 周期倍数 P' の探索
  (a) $T = x_0, H = M(T) = x_1, P' = 1$ とする
  (b) $T = H$ となるまで，以下の処理を繰り返す：
    $T \leftarrow M(T), H \leftarrow M(M(H)), P' \leftarrow P' + 1$
(2) リンク長 L の特定
  (a) $H = M(T) = x_{P'}$ を計算し $T = x_0, L = 0$ とする
  (b) $T = H$ となるまで，以下の処理を繰り返す：
    $T \leftarrow M(T), H \leftarrow M(H), L \leftarrow L + 1$
(3) 周期 P の特定
  (a) $H = M(T) = x_{L+1}$ を計算し $P = 1$ とする
  (b) $T = H$ となるまで，以下の処理を繰り返す：
    $H \leftarrow M(H), P \leftarrow P + 1$

実際に以下の系列 S を用いて Floyd 法によって周期 P とリンク長 L を求める様子を図 1 に示す．

$$S = (5, 18, 51, 41, 58, 21, 56, 28, \underline{63, 3, 11, 36}, \underline{63, 3, 11, 36}, \cdots)$$

なお，下線部で示した部分系列がループに相当し，周期 $P = 4$, リンク長 $L = 8$ となる．この例では，図 1 の最上段に示したように，周期倍数 $P' = 12$ となっているが実際の P は 4 である．このように Floyd 法はアルゴリズムを完走しないと周期 P が判明しない．

2.4　Brent 法

Brent 法 [2] は，Floyd 法をより効率良く周期を探索するためのアルゴリズムとして提案された．文献 [1] でも簡潔に紹介されている．

Brent 法の特徴は，記憶しておく要素数が Floyd 法の 2

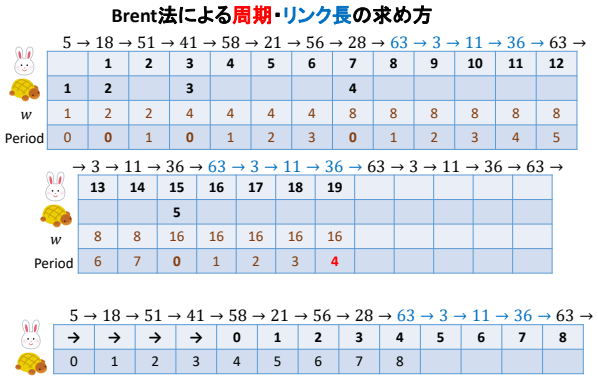

図 2　Brent 法による周期リンク長探索例

個に加え累乗数を記憶しておく変数 w が必要である．また，最初に行うメインループにて，周期 P の値を確定することができる．そのため，周期のみが必要でリンク長は計算する必要がない場合はその時点で計算を打ち切ることも可能である．文献 [2] によれば，Floyd 法よりも最大 36% 高速であると示されている．

以下に Brent 法のアルゴリズムを示す．記憶しておく要素を T と H とする．また初期値 $IV = x_0$ とする．

(1) メインループ：　周期 P の探索
  (a) $T = x_0, H = M(T) = x_1, P = 1, w = 1$ とする
  (b) $T = H$ となるまで，以下の処理を繰り返す
    (i) もし $l = w$ ならば，$T \leftarrow H, P \leftarrow 0, w \leftarrow 2w$
    (ii) $H \leftarrow M(T), P \leftarrow P + 1$
(2) リンク長 L の特定
  (a) $H = T = x_0, L = 0$ とする
  (b) P 回 $H \leftarrow M(H)$ を繰り返す
  (c) $T = H$ となるまで，以下の処理を繰り返す:
    $T \leftarrow M(T), H \leftarrow M(H), L \leftarrow L + 1$

Floyd 法と同じ系列 S を用いて Brent 法によって周期 P とリンク長 L を求める様子を図 2 に示す．この例では，図 2 の最上段に示したように，H は順番に写像を繰り返していくが，T は間隔を 2 倍ずつ広げながら途中で H の値を上書きしていく．その上書きする度に P の値をリセットしそこから同じ値が出現するかどうかを探索していく．この場合，H の値が途中の写像値を飛ばすことなく推移していくため，P は必ず周期となる．

3.　様々な提案アルゴリズム

ここではオリジナルの Floyd 法および Brent 法における写像回数を算出し，これを元に幾つかの改良版アルゴリズムを提案する．

3.1　Floyd 法と Brent 法の写像回数

Floyd 法，および Brent 法に共通することは，どちらも確率的な試行が無く確定的なアルゴリズムであるという点である．これは，与えられた系列の周期 P とリンク長 L に

よって，必要となる写像回数が決定することを意味している．これらのアルゴリズム上での写像回数について，以下に 2 つの定理を示す．

[定理 1]

周期 P とリンク長 L である系列に対する Floyd 法の写像回数は，$4P + 5L - 3(L \bmod P) - 1$ である．

[証明]

Floyd 法の手順 (1) で求める周期倍数 P' について，整数 m を使って $P' = mP$ と表す．H は $(x_1, x_3, \cdots, x_{2i-1} \cdots)$ と変化していくのに対して，T は $(x_0, x_1, \cdots, x_{i-1}, \cdots)$ のように変化する．ここで初めて $H = T$ となるのを $H = x_{2P'-1}, T = x_{P'-1}$ と表す．L と P の大小関係で，この時の m の値が決まる．

$L < P$ ならば，$i = P, 2i = 2P$ のときに初めて $x_{i-1} = x_{2i-1}$ を満たす．このとき $m = 1$ である．

$P \leq L < 2P$ ならば，$i = 2P, 2i = 4P$ のときに初めて $x_{i-1} = x_{2i-1}$ を満たす．このとき $m = 2$ である．

これを繰り返していくと，$m = \lfloor L/P \rfloor + 1$ であることがわかる．ここで，$\lfloor L/P \rfloor P = L - (L \bmod P)$ だから，$P' = mP = L - (L \bmod P) + P$．$P'$ の値が分かったので，H と T が実際に写像した回数がこれからわかる．H は x_0 から $x_{2P'-1}$ まで，T は x_0 から $x_{P'-1}$ までそれぞれ写像するから，その合計回数は，$(3P' - 2) = (3P + 3L - 3(L \bmod P) - 2)$ となる．

次に，手順 (2) では，H は $x_{P'-1}$ から $x_{P'-1+L}$ まで $(L + 1)$ 回，T は x_0 から x_L まで L 回写像を行うから，合計で $(2L + 1)$ 回写像を行う．

最後に手順 (3) で，T は x_L から x_{L+P} まで写像するが H は写像しない．よって P 回写像を行う．

これらの合計から，全ての写像演算回数は，

$$\begin{aligned} &(3P + 3L - 3(L \bmod P) - 2) + (2L + 1) + P \\ &= 4P + 5L - 3(L \bmod P) - 1 \end{aligned}$$

□

[定理 2]

周期 P とリンク長 L である系列に対する Brent 法の写像回数は，$2(P + L) + 2^E - 1$ である．ただし，E は $2^{E-1} \leq \max(P, L + 1) < 2^E$ を満たす整数とする．

[証明]

Brent 法における H は，x_0 から順番に写像を繰り返していく．これに対して T は以下のような飛び飛びの値が出現する．

$$T : (x_0, x_1, x_3, x_7, \cdots, x_{2^k-1} \cdots)$$

周期 P の探索が終了する場合の条件を考えると，T がループ内に入っていること，H が T から丁度 1 周期分離れた位置にいることである．よって，周期が判明したときの k を E とすると，$(2^E - 1)$ はリンク上にないので L よりも大きい，つまり $2^E > (L + 1)$ である．また，続く 2^E 個の要素内で 1 周期分が含まれているから $2^E > P$ も満たす．よって，このような条件を満たす最小の E は $2^{E-1} \leq \max(P, L + 1) < 2^E$ となる．

これから，周期探索終了時の T は $T = x_{2^E-1}$ であるので，H はそれより 1 周期分先の $H = x_{P+2^E-1}$ である．T は写像演算せず，H のみが x_0 からこの値まで写像演算を行うから，周期探索の写像回数は $P + 2^E - 1$ となる．

次にリンク長 L の特定である．予め H は x_0 から P 回写像演算を行い，x_P としておく．T を x_0 に戻し，ここから H と T をそれぞれ L 回写像する．よって，この手順で必要な写像回数は $(P + 2L)$ である．

これらの合計から，Brent 法における写像回数は，$2(P + L) + 2^E - 1$ である． □

定理 1, 定理 2 から，Floyd 法と Brent 法における写像演算回数は，周期 P やリンク長 L の高々数倍程度に抑えられていることがわかる．特に，要素が有限で最大値 E_{max} よりも小さな要素 $(0 \leq X < E_{max})$ で構成された系列の周期・リンク長を求める場合は，写像回数の上限を求めることができる．

ここで，U を周期 P とリンク長 L の和とする．この U は，系列 S の初期値 x_0 から全て異なる要素が得られる長さを示しており，長さ U の部分系列 $(x_0, x_1, \cdots, x_{U-1})$ は全ての要素が互いに異なる．U はユニークな系列長を示している．

この U によって，Floyd 法の写像回数は高々 $5U$，Brent 法の写像回数も高々 $3U$ だとわかる．また，この U は M 以下であることは自明である．なぜなら，$0 \leq X < E_{max}$ を満たす相異なる要素は E_{max} 個しかないからである．よって，与えられた系列の要素が E_{max} で制約を受ける有限要素であるならば，Floyd 法，Brent 法の写像回数はそれぞれ $5E_{max}$, $3E_{max}$ よりも小さい．

3.2 写像値記録版

Floyd 法，Brent 法の改良として，まず最初に両者の写像値記録版を提案する．これは，それぞれのアルゴリズムで演算した途中の写像値を全てメモリ上に記憶しておくことで，2 回目以降の写像演算を行わずに済ませる方式である．

第 3.1 節より，各アルゴリズム内のそれぞれの手順における写像回数を解析した．ここで，両方のアルゴリズムに共通する点として，どちらも最初の H が写像した要素を全てメモリ上に記憶しておけば，他の写像演算は必要なくそのメモリ上の要素を参照するだけで演算可能であることが挙げられる．図 1 と図 2 の例では，図 1 では H が最初に x_0 から x_{31} まで 31 回写像演算を行う．この途中を含めた全ての写像値 $(x_0, x_1, \cdots, x_{31})$ を記録しておけば，それ

以降の写像値計算は全て不要でこの記憶領域から参照することで代替することができる．同じように，図 2 でも，H が最初に計算する x_0 から x_{19} までの写像演算結果を全て記録しておくことで，それ以降の全ての写像演算を省略できる．特に，これらの記録を計算機の主記憶装置に保存しておくことができれば，高速に読み書きすることができるので処理の高速化を行うことができる．

ただし，写像値記録版を実行する場合はその要素を全て記憶できるメモリが確保できることが条件となる．定理 1 と定理 2 から，通常版と写像値記録版それぞれでの写像回数と必要なメモリ量（要素サイズを 1 とした値) を以下の表 1 に示す．ここで，U を用いると，写像値記録版の

表 1 Floyd 法,Brent 法の通常版と写像値記録版それぞれの写像回数と必要メモリ量

アルゴリズム	写像回数	メモリ量
通常版 Floyd 法	$4P+5L$ $-3(L \bmod P)-1$	2
通常版 Brent 法	$2(P+L)$ $+2^E-1$	2
写像値記録版 Floyd 法	$2P+2L$ $-2(L \bmod P)-1$	$2P+2L$ $-2(L \bmod P)-1$
写像値記録版 Brent 法	$P+2^E-1$	$P+2^E-1$

Floyd 法の写像回数と必要メモリ量は $2U$ 未満である．また，$P \geq 1$ で $2^{E-1} < \max(P, L+1)$ より $2^{E-1} < P$ かつ $2^{E-1} < (L+1)$ である．これらから $2^E < (P+L+1)$ であるから，$2^E - 1 < (P+L) = U$ となる．よって，写像値記録版の Brent 法の写像回数と必要メモリ量も $(P+U) < 2U$ 未満である．写像回数は，通常版の $5U, 3U$ からそれぞれ $2U$ に減少している．

これから，例えば全ての要素が 2^{32} ビットの整数であるような系列であれば，その周期・リンク長探索に必要なメモリ量は，$2U = 2^{33}$ 個の要素があれば良い．1 要素あたり 4bytes 必要であるから，全体では 4×2^{33}bytes，つまり 32GB のメモリを確保できれば実行することができる[*1]．よって，本稿執筆時の計算機性能より，要素の種類が 2^{32} 未満であれば写像値記録版で周期・リンク長探査が実行可能である．

3.3 併用版 Brent アルゴリズム

第 3.2 節で示したように写像値記録版では要素種類に上限があるものの，写像演算を減らしてより高速に演算することが可能である．また，表 1 より，Floyd 法よりも Brent 法の方が通常版・写像値記録版どちらもより少ない写像回数で周期・リンク長探索を行うことができる．よって，これ以降は Brent 法に絞って更なる改良版を提案する．

まずは，併用版 Brent アルゴリズムである．これは，写像値記録版 Brent アルゴリズムの欠点であるメモリ量の制限について，写像値記録で処理できるものは全て写像値記録で行い，それを超えた範囲では通常版の Brent 法で計算するものである．途中の計算も，メモリ上に記録されている写像値は全てそちらを用いて，それを超える部分はその都度写像をし直す．

併用法の利点は，写像値記録版のようなメモリ量の上限を超えたものに対しても周期探索を行うことができ，メモリ量の上限範囲であれば写像値記録版に近い高速な処理を行うことができることである．また欠点としては，両方の構造を含むためアルゴリズムが複雑であり，写像値記録状態か通常状態かの条件分岐が入るため全てメモリ上で処理できた場合でも写像値記録版ほどの高速な演算を行えないことである．

3.4 3 倍版 Brent アルゴリズム

第 2.4 節で示したように，Brent 法では T の更新間隔を決める変数 w を 2 倍 2 倍と増加していく．しかし，この間隔は必ず 2 倍である必要はない．文献 [1] によれば，$w \leftarrow (\lfloor 2.4771w \rfloor + 1)$ というものが最も効率が良いと記載されているが，このような小数の乗算や端数切捨てに必要な除算処理を考えるとこれが必ずしも高速化に繋がるとは言えない．そこで，2 倍でなく 3 倍に変更する．これによって，除算を増やすこと無く T の更新間隔を広げることができる．

3 倍にした利点は，T の更新間隔を広げることで比較的写像回数が低い部分で T の更新回数を削減することができる．欠点は，より間隔が広がることで写像値記録版に必要な最大メモリ量が $2U$ から $3U$ に増加することである．

3.5 性能比較

これまで説明してきた Floyd 法，Brent 法とそれらの派生アルゴリズムについて，実際に系列の周期・リンク長を探索してその計算時間を比較する．

まずは，通常版と写像値記録版の比較である．それぞれの方式で以下の条件を満たす系列の周期・リンク長を全て計算し，その時間を測定した．また，通常版 Floyd 法と，写像値記録版 Floyd 法をそれぞれ 1 とした場合の速度比を算出しそれらの結果を表 2 に示す．

- 使用した写像：　素体上のロジスティック写像 [5], [6]

$$\mathrm{LM}_{\mathbf{Z}_p[\mu_p]}(X) = \mu_p X(X+1) \bmod p$$

- 法 $p = 64007$(固定)
- コントロールパラメータ μ_p：1 から $(p-1)$ まで全て
- 初期値 IV：　各 μ_p 毎に 1024 個ランダムに選択[*2]

[*1] 実際には，より小さなメモリ量で実行可能だが，周期・リンク長が未知の系列を探索する場合は考えられる最大のメモリ量を確保しておかなければ途中で計算が停止する可能性がある

[*2] ただし，アルゴリズム間での初期値選択に差が生じないように，

つまり，各アルゴリズム毎に $64006 \times 1024 = 65542144$ 個の組 (μ_p, IV) から生成される系列の周期・リンク長を計算した時の演算時間である．測定に使用した PC のスペックを以下にまとめる．

- CPU：Intel Core i7-4770 (3.40GHz)
- メインメモリ：32GB
- OS：CentOS6.10

表 2 Floyd 法,Brent 法の通常版と写像値記録版の計算時間と速度比（左：通常 Floyd 法=1, 右：写像値記録 Floyd 法=1)

アルゴリズム	演算時間	速度比	
通常版 Floyd 法	26 分 40.0 秒	1.000	1.963
通常版 Brent 法	23 分 46.6 秒	0.892	1.751
写像値記録版 Floyd 法	13 分 34.9 秒	0.509	1.000
写像値記録版 Brent 法	13 分 13.3 秒	0.496	0.973

この結果から，通常版と写像値記録版は Floyd 法，Brent 法どちらも約 2 倍写像値記録版の法が高速に周期探索を行うことができる．また，Floyd 法と Brent 法については，通常版では Floyd 法に比べて通常版は約 1 割高速化できているが，写像値記録版では両者の差は 3%以下である．

次に，併用版 Brent 法，3 倍 Brent 法について通常版，写像値記録版の Brent 法と比較した．その結果を表 3 に示す．なお，写像と初期パラメータ選択，試行した探索回数などは全て表 2 と同じものである．この結果から，3 倍

表 3 素体上ロジスティック写像での各 Brent 法の計算時間と速度比（左：通常版=1, 右：写像値記録版=1)

アルゴリズム	演算時間	速度比	
通常版 Brent 法	23 分 46.6 秒	1.000	1.798
通常版 3 倍 Brent 法	22 分 33.5 秒	0.949	1.706
写像値記録版 Brent 法	13 分 13.3 秒	0.556	1.000
写像値記録版 3 倍 Brent 法	13 分 05.3 秒	0.550	0.990

Brent 法は通常版で約 5%，写像値記録版でも約 1%ほどの高速化を実現している．ただし，写像値記録版では必要となるメモリ量が 1.5 倍となるため，そこまでのコストをかけている割にはそこまでの高速化とも言えない．よって，3 倍版を使用する場合は，通常版 Brent 法で利用すべきである．

次に，写像と演算精度を変化して，写像値記録版で実行できるものとできないもので速度比を計算する．

- 使用した写像：　整数上のロジスティック写像 [3], [4]

$$\mathrm{LM}_{\mathrm{Int}}^{(n)}(X) = \left\lfloor \frac{\mu X(2^n - X)}{2^n} \right\rfloor$$

- 演算精度 n : $20, 24, 28, 30, 32, 36, 40, 44, 48, 52, 56, 60$
- コントロールパラメータ[*3] μ : $3.875 < \mu \leq 4$

$$\mu = \frac{4 \times 2^{n+3} - M}{2^{n+3}},\ M = 0, 1, 2, \cdots, 2^n - 1$$

- 初期値 IV : 0 から $(2^n - 1)$ までの任意の値
- 探索回数：(μ, IV) の組を，2^{20} 個 $(n \leq 32)$ / 2^{10} 個 $(n \geq 36)$ ランダムに選択

なお，使用した計算機の都合上，写像値記録版は $n = 30$ 以下のデータのみ計算可能であった．この探索時間を表 4 にまとめる．また，通常版 Brent 法を 1 とした各 Brent 法の速度比を図 3 に示す． この表から，演算精度 n が写像値記

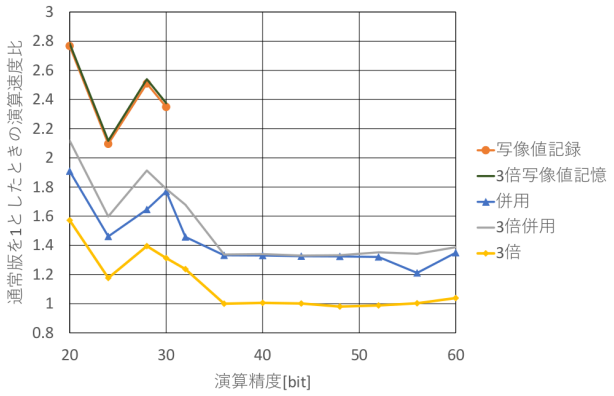

図 3 複数の Brent 法による整数上のロジスティック写像での系列周期探索速度比

録版を利用できる $n < 32$ の範囲では各 Brent 法で 3 倍演算を行うと効率良く周期探索を行うことができるが，演算精度が上がるにつれて 3 倍演算の利点が減少してくる．これは，ある程度までは 3 倍演算によって T の更新を行う間隔を広げる効果が期待できたものの，演算精度が大きくなるとその差があまり影響してこないことを意味している．また，通常版に対する併用版の利得も，演算精度が大きくなるとその比率が下がっていき，ほぼ 1.3 倍程度に収まる．

併せて，1 回の周期探索時間についても調査する．表 5 は，通常法と併用法，それぞれの 3 倍版の 4 種類の方式について，$n \geq 44$ の範囲で平均周期探索時間を示している．これより，演算精度 $n = 60$ であっても通常 Brent 法で 1 つの系列に対して平均で約 10.18 秒，併用版であれば約 7.54 秒程度で周期の探索を行うことができる．この演算精度は，得られた系列を擬似乱数系列として NIST 検定に全て合格できる程度の乱数性を持っている．そのような

全てのアルゴリズムで同じ初期値が選ばれるように乱数種を設定している

[*3] コントロールパラメータの定義は幾つか存在するが，ここでは [7] のサブコントロールパラメータ M を用いた方式を採用した

表 4　整数上ロジスティック写像での各 Brent 法の計算時間

演算精度	探索回数	演算時間 [秒]					
		通常	写像値記録	写像値記録 3 倍	併用	併用 3 倍	3 倍
20	1048576	27.20	9.83	9.77	14.25	12.84	17.30
24		73.80	35.21	34.86	50.50	46.18	62.72
28		325.67	129.76	128.27	197.87	170.27	233.47
30		590.43	251.36	248.52	333.65	330.31	449.79
32		1069.65	演算不可	演算不可	733.79	637.29	865.01
36	1024	3.25			2.44	2.43	3.25
40		12.48			9.38	9.31	12.41
44		49.21			37.12	37.00	49.13
48		182.56			137.91	137.04	186.21
52		676.99			512.58	500.58	684.99
56		2660.82			2197.23	1982.35	2653.55
60		10420.58			7721.18	7512.40	10030.75

表 5　整数上ロジスティック写像による生成系列における演算精度に対する各 Brent 法の 1 回あたりの周期探索平均時間

演算精度	平均探索時間 [秒]			
	通常	併用	併用 3 倍	3 倍
44	0.048	0.036	0.036	0.048
48	0.18	0.13	0.13	0.18
52	0.66	0.50	0.49	0.67
56	2.60	2.15	1.94	2.59
60	10.18	7.54	7.34	9.80

系列でも，数秒程度で周期の特定が可能であることを示している．これは，擬似乱数生成器の初期値選択において，ランダムに選んだ初期値が偶然小さな周期を持つ系列を生成し，得られる擬似乱数の乱数性が低くなることを事前に検査するのに十分利用できることを意味している．

4. 最適な周期探索アルゴリズムの選択

本稿では，周期・リンク長が未知の系列に対する周期・リンク長探索アルゴリズムとして，Floyd 法と Brent 法を調査し，そのアルゴリズムから写像値記録版や併用版，3 倍 Brent 法などを提案し，それらの実行条件と演算速度を比較した．調査した範囲では，通常版，写像値記録版ともに Brent 法の方が Floyd 法よりも高速に動作する．あとは，この通常版，写像値記録版，併用版，3 倍版と 4 種類ある Brent 法を，どのように選択するのかの指針が必要である．

周期の探索を行う場合，幾つかの状況が考えられる．

(1) 単体，もしくは高々数個の系列の探索
- 小さな周期のループに到達しないかどうかの判定
- 理論的解析結果の周期理論値と実測値との比較
- 系列生成プログラムの動作テスト

(2) 多くの初期パラメータを用いた複数の系列の探索
- 最適な初期条件の探索
- 全初期値からの平均周期・リンク長算出

状況 (1) に関しては，メモリを多く必要としたとしても高速に計算できる写像値記録版，併用版 Brent 法を用いるべきである．また，状況 (2) に関しては，逆にメモリ使用量は減らして CPU コア単体での演算速度は半減するが，複数コア・複数計算機を用いて並列計算を行うのに最適な通常版・3 倍版 Brent 法を用いるべきである．よって，本稿における周期探索状況毎の周期探索アルゴリズム選択指針としては，表 6 のようにまとめることができる．

表 6　周期探索アルゴリズムの選択指針

演算精度 \ 試行数	単体・数回	多数・全初期値
32bit 未満	写像値記録版 Brent 法	3 倍 Brent 法
32bit 以上	併用版 Brent 法	通常版 Brent 法

5. まとめ

本稿では，周期・リンク長探索アルゴリズムである Floyd 法，および Brent 法について調査した．それらの方式で必要な写像回数を算出し，これに基づき写像値記録版の両手法での動作に必要なメモリ量について調査した．その結果，32 ビット整数値を要素とする範囲であれば写像値記録版でも動作できることが判明し，数値実験から写像値記録版は通常版の約 2 倍程度高速に動作することが判明した．

また，Brent 法については，通常版と写像値記録版を組み合わせた併用版，2 倍ずつ更新する範囲を 3 倍に変更した 3 倍 Brent 法なども提案し，それらの実行時間について調査した．その結果，併用版は通常版よりもおよそ 1.3 倍程度高速に演算できること，3 倍版は 30bit 以下の比較的低演算精度の範囲では高速化に寄与するが，それを超えると通常版とあまり変化がないことがわかった．

これらの結果から，周期の探索を行うときの状況に合わせて最適な周期探索アルゴリズムを選択する選択指針を示した．演算精度と探索する系列の個数によって 4 通りに分け，それぞれ最も効率的なアルゴリズムの選択が行えることを示した．

今後は，Brent 法以外にもより効率的な周期・リンク長

探索アルゴリズムがあるのかどうかを調査し，今回の様々な改良版 Brent 法と比較していきたい．また，周期・リンク長の完全な特定ではなく，ある範囲に入る，もしくはある値以下（以上）であることの判定など，完全に周期・リンク長を特定しなくても良い条件でさらに高速な探索ができないかを考えていきたい．

参考文献

[1] Knuth, D. E.: *The Art of Computer Programming*, Third Edition, pp.7, 10-40, 539, fundamental algorithms (1997).
[2] Brent, R. P.: *An improved Monte Carlo factorization algorithm*, BIT Numerical Mathematics , 20 (2): pp. 176184, (1980).
[3] T. Miyazaki, S. Araki, and S. Uehara, “Some Properties of Logistic Maps over Integers,” *Special Section on Signal Design and its Application in Communications, IEICE Trans. Fundamentals*, Vol.E93-A, No.11, pp.2258-2265, 2010.
[4] T. Miyazaki, S. Araki, Y. Nogami, and S. Uehara, “Rounding Logistic Maps over Integers and the Properties of the Generated Sequences,” to appear in IEICE Trans. Fundamentals, Vol. E94-A, No.9, pp.1817-1825, 2011.
[5] 宮崎武, 荒木俊輔, 上原聡, 野上保之, “素体上のロジスティック写像による擬似乱数生成に関する一考察,” 2013 年暗号と情報セキュリティシンポジウム (SCIS2013), pp.1-7, 2013.
[6] T. Miyazaki, S. Araki, S. Uehara, and Y. Nogami, “A Study of an Automorphism on the Logistic Maps over Prime Fields,” Proc. of The 2014 International Symposium on Information Theory and its Applications (ISITA2014), pp.727-731, 2014.
[7] 荒木俊輔, “擬似乱数生成器に用いる整数上のロジスティック写像の最小周期の保証に向けた手法に関する一考察,” 第 5 回有限体理論とその擬似乱数系列生成への応用ワークショップ, 2019.