三重大学教育学部研究紀要　第56巻　自然科学　(2005)　59－63頁

実数シフト乱数生成法のメッセージダイジェスト性について

谷口礼偉*

The Message-Digest Property of the Shift-Real Random Number Generator

Hirotake YAGUCHI

要　旨

最近考案された非代数的非再帰的擬似乱数生成法である実数シフト法のアルゴリズムを利用して、新しくメッセージダイジェスト関数が作成可能であることを、関数出力の統計的検証を通じて示す。

SR法によるメッセージダイジェストの概要

メッセージダイジェスト関数（MD関数）は、任意長の入力メッセージに対し、その要約を出力する関数である。一般に、その出力は固定長であり、一様乱数性を要求されることが多い。ここで一様乱数性とは、一連のMD関数の出力値を並べていくと、どの値もまったくでたらめにかつ同程度に現れるということである。また、情報セキュリティの観点からは、出力するデータ長を長く取って（128あるいは160ビット以上)、同じ出力値をもつ異なる入力メッセージが見つけにくい、という性質も要求されている。これらのことを考慮すると、MD関数を、1つの入力メッセージに対して1つの乱数値を出力する擬似乱数発生器としてとらえることができる。MD関数はハッシュ関数ともいわれ、著名なものとしてMD5, SHA-1, RIPEMD-160　などがある([4])。

我々は、[1, 2, 3] で非再帰的・非代数的な擬似乱数生成法として、実数シフト法（the Shift-Real method, SR法）を提唱した。本論文の目的は、このSR法のアルゴリズムを利用して、新たにメッセージダイジェスト関数が作成可能であることを示すことである。

実数シフト法は、次の単純実数シフト計算（the simplified shift-real computation, SSR計算）が基本である:

$$g(x) = \underbrace{x \cdot x \cdot x \cdot \cdots \cdot x \cdot x}_{24}, \quad x \in [1,\ 2)$$

を

$$u_0 := 1 \qquad u_k := u_{k-1} \times x, \quad k = 1, 2, \cdots, 24,$$

と倍精度計算し、u_k を1回計算するごとに、u_k を表す倍精度変数の

i）仮数部の全てのビット値を1ビット左にシフトし、
ii）指数部は　$\cdots \times 2^0$ となるように設定する。

* 三重大学教育学部数学教室

また、実際の乱数値としては、x を変化させ、対応する計算結果 u_{24} から、上位 3 桁を棄て続く 4 桁を取り出すものであった。我々は、この実数シフト計算中の (i) の左シフトを行った結果生じる右端ビットの空白に注目する。すなわち、上述の SSR 計算ではこの右端ビットに 0 を入れているが、我々は経験上ここに何を入れても乱数特性があまり変わらないことを知っている。よってシフトのたびごとに生じるこの空いたビットに、予め与えられている入力メッセージを構成するバイト列のビットを順次埋め込んでいくことを繰り返せば、入力メッセージの情報が埋め込まれた形の倍精度変数値 u が得られる。この u を新たに x と考えて改めて SSR 計算を行えば、入力メッセージの情報が含まれた乱数値が得られることになる。このようにして倍精度変数を使った SSR 計算で一様乱数が得られることが分かれば、同様のアルゴリズムを、多倍長の整数計算で行うことにより、例えば 160 ビットの出力を持つメッセージダイジェスト関数を実現することも可能である。本論文では、この意味で、出力ビットは少ないものの、今までの我々が使用した乱数検定プログラムがそのまま使える SSR 計算に基づいた MD 関数に対象を絞って、一様乱数性の検定を中心に考察を進めていくことにする。

メッセージダイジェストのアルゴリズム

SR 法のアルゴリズムに基づいた MD 関数作成のアイデアは上述のとおりであるが、実際の計算においては、1 ビットずつ扱うのでは能率が悪いので、一度に 1 バイトの入力データ情報を埋め込むなどの工夫を行うことになる。また、埋め込み過程で使われる乗数 x を固定したままでは特性が偏るので、定期的に x の値を変える必要がある。SR 法のアルゴリズムを用いた MD 関数の具体的なアルゴリズムは以下のとおりである。

変数

SSR 計算の x に相当する倍精度変数を X とする。

この X には、[3] に述べられている関数 nextX() により、値が順次セットされていく。

SSR 計算の u に相当する倍精度変数を U とする。

この U に、入力メッセージの情報が埋め込まれていくことになる。

変数の初期化

X を　SRIni(0.0);　X＝nextX(); により初期化する（[3] 参照)。

U には　1.2718281828459　を入れておく。

入力メッセージの埋め込み

(1) U の仮数部を 8 ビット左シフトする。

(2) 入力メッセージの 1 バイトを読み、U の仮数部の b_{25} ～b_{32} と XOR をとる。

(3) X＊U を作り、U に保存する。

(4) U の指数部を $\times 2^0$ にする。

埋め込みプロセスの繰り返し

上述 (1) ～ (4) を、入力データが無くなるまで繰り返す。ただし、X を 8 回使うごとに（＝入力データを 8 バイト処理するごとに)、nextX() により X を更新する。

メッセージダイジェストの作成

最終的に得たUを、SSR計算のXとしてK改良SSR計算を行う。ただし繰り返し回数は64回とする。得た値の最初の5桁を棄て続く8桁をメッセージダイジェストとして採用し、SSR-MD値ということにする。

メッセージダイジェスト値の一様乱数性の検証

乱数の特性を検証するには、多量の乱数が必要である。このため、整数nを4バイトで表し、これを1つの入力メッセージMSGnとする。MSGnに対して、10進8桁のSSR-MD値$d_1\,d_2\,d_3\,d_4\,d_5\,d_6\,d_7\,d_8$を求め、この乱数特性を[3]で使われた方法で検定する。具体的には、8桁MD値 を上位4桁$d_1\,d_2\,d_3\,d_4$、下位4桁$d_5\,d_6\,d_7\,d_8$の2つに分割し、それぞれについて20000個の乱数値に対する次の7種類計10の検定を危険率0.05で行う（詳細は[3]を参照のこと）：

[検定Ⅱ] 文字0〜9の出現頻度のχ^2検定、
[検定Ⅲ] 文字0の出現間隔のχ^2検定、
[検定Ⅳ] 乱数値のKolmogorov-Smirnov検定（K^+およびK^-）、
[検定Ⅴ] 単純上昇連および下降連テスト、
[検定Ⅵ] 4枚の0〜9カードによる古典ポーカーテスト、
[検定Ⅶ] 遅れ1および2の系列相関テスト、
[検定Ⅷ] 衝突テスト。

そして仮説が棄却された回数c（$0 \leq c \leq 10$）を数える。この操作を1000回繰り返す。各検定が危険率0.05で独立に行われると仮定すれば（註: この仮定は厳密には正しくない），cの分布は二項分布Bin(10, 0.05)になるので、実際に得られたcの分布と対照してχ^2検定値（CHITEST）を求める。

結果は、以下のようになる。

cの値 ⇒	0	1	2	≥ 3	CHITEST
上位4桁	607	307	76	10	0.908899
下位4桁	602	305	84	9	0.559416
Bin(10, 0.05)	598.7	315.1	74.6	11.5	

危険率0.05に比してCHITESTの値はかなり良い結果となっている。また、各検定の棄却回数は次のとおりである:

検定 ⇒	(Ⅱ)	(Ⅲ)	(Ⅳ)	(Ⅳ)	(Ⅴ)	(Ⅴ)	(Ⅵ)	(Ⅶ)	(Ⅶ)	(Ⅷ)	計
上位4桁	60	51	42	47	48	56	55	45	40	47	491
下位4桁	48	51	42	49	54	58	59	52	41	46	500

危険率0.05の検定を1000回行った結果であるから、各検定とも妥当な値を示しているといえよう。

上述の検定では、入力メッセージがすべて4バイトであったが、MD関数の実際の入力メッセージはいろいろな長さを持っている。我々の身近にある多量のメッセージ源としてはコンピュータのファイルがあるので、我々が使用しているパソコン（Winodows Me）の全ファイルを入力メッセージとして、SSR-MD値の特性を調べてみた。入力メッセージ数（全ファイル数）は30320であり、このうち入力

メッセージ長が同じで、対応するSSR-MD値も同じになる（重なる）ファイル数は3974であった。これらは、同一ファイルが別のディレクトリにあったり、名前を変えて保存されているものと思われる。これらを除いた26346個のファイルについて、入力メッセージ長が異なるが、SSR-MD値が重なるファイル数は5であった（重なりの度合いはすべて2)。これらは偶然にSSR-MD値が同じになったものと思われる。この重なり数が妥当なものであるかどうかの見当をつけるために、衝突回数の理論確率およびBorlandのフリー版Cコンパイラー bcc32で8桁の乱数（4桁乱数を2つ結合したもの）を26346個作成し、その衝突回数（乱数の重なり回数）を調べることを1000回繰り返したときの、衝突回数の分布を見てみると次のようになる。

衝突回数	0	1	2	3	4	5	6	7以上
bcc割合	.030	.111	.186	.209	.193	.134	.080	.057
理論確率	.031	.108	.187	.217	.188	.131	.075	.063

これから見ると5という値は、やや多いが特に問題とする値ではない。

[検定Ⅱ]～[検定Ⅷ]の7種類10検定を、先ほど得た26346個の10進8桁SSR-MD値のうちの最初の20000個に対して適用すると以下のようになる。

検定 ⇒	(Ⅱ) (>0.05)	(Ⅲ) (>0.05)	(Ⅳ) (<1.2239)	(Ⅳ)
上位4桁	0.7652	0.2939	0.7354	0.891
下位4桁	0.2642	0.996	1.718Δ	0.3394

検定 ⇒	(Ⅴ)	(Ⅴ) (>0.05)	(Ⅵ) (>0.05)
上位4桁	0.0588	0.8069	0.1541
下位4桁	0.933	0.639	0.02599Δ

検定 ⇒	(Ⅶ)	(Ⅶ) ([−0.01419, 0.01409])	(Ⅷ) (<62)
上位4桁	0.009878	−0.001954	60
下位4桁	−0.001655	0.0000635	47

（検定番号の下の（ ）内の数値は、危険率を0.05とした場合の仮説が棄却されない範囲である。）各検定を1回行っただけの結果であることを考えると、妥当なものといえよう。

まとめ

実数シフト法のアルゴリズムを利用した上述のメッセージダイジェスト関数は、検定結果を見る限り、一連の入力メッセージに対して一様乱数を生成しているものと判断される。したがって、今後、アルゴリズム中に使われている不動小数点演算をすべて多倍長の整数演算に置きかえることにより、十分な出力データ長をもつメッセージダイジェスト関数が作成可能であると考えられる。同時に、情報セキュリティの観点から、このメッセージダイジェスト関数の安全性についても理論的な考察を高めておくことが必要である。

参考文献

[1] Yaguchi, H.: Randomness of Horner's rule and a new method of generating random numbers. Monte Carlo Methods and Appl., Vol. 6 (2000), 61－76.

[2] Yaguchi, H.: Construction of a long-period nonalgebraic and nonrecursive pseudorandom number generator. Monte Carlo Methods and Appl., Vol. 8 (2002), 203－213.

[3] 谷口礼偉: 数値計算誤差と乱数生成. 神戸大学理学部数学教室、2001.

[4] 暗号技術検討会: 2001 年度報告書. 経済産業省・総務省、2002.