アルゴリズム 55－1
(1997. 1. 23)

単純指標を持つ乱順列の高速生成法

二村良彦 † 大谷啓記 † 青木健一 † 二村夏彦 ‡

† 早稲田大学 理工学部 情報学科
‡ School of Computer and Information Science, Syracuse University

長さ n の順列を等確率，即ち $1/n!$ で生成する $O(n)$ アルゴリズムは知られている．しかし，整列アルゴリズム等の精密な評価のためには，このような一様乱順列による評価では不十分である．例えば，一様乱順列に含まれる葉数 (自分よりも小さい隣人を持たない要素の個数)，ランズ，および上昇部分の個数 (即ち n–ランズ) は，平均各々約 $(n+1)/3$，$(n+1)/2$，および $(n-1)/2$ である．これは一様乱順列が極めて偏った特性を有することを意味する．アルゴリズムの性能に影響を及ぼす性質 (例えば葉数) を制御しながらランダムに順列を生成し，それを用いてアルゴリズムの性能を評価する必要がある．本稿では，順列から非負の整数上への関数及び関数値を順列の特性指標と呼ぶ．特性指標の中でも，順列の葉数，ランズ，上昇部分数等に対応するクラスを単純指標と呼び，それを形式的に定義する．そして長さ n，単純指標 m を持つ乱順列を $O(nm)$ で生成する計算機オーバーフロー (またはアンダーフロー) 無しの方法を提案する．また，単純指標が葉数である場合には順列を $O(n)$ で生成する実用的近似方式について報告する．

Fast Generation of Random Permutations with Simple Indexes

†Yoshihiko FUTAMURA †Hirofusa OTANI
†Ken-ichi AOKI ‡Natsuhiko FUTAMURA

† School of Science and Engneering, Waseda University
‡ School of Computer and Information Science, Syracuse University

$O(n)$ algorithms for uniformly generating random permutations are well known. However, in order to evaluate performance of algorithms, uniform random permutations are insufficient because they are not fair in some sense. The reasons, for example, are that they have approximately $(n+1)/3$ leaves (i.e. elements in a permutation which do not have smaller neighbors), $(n+1)/2$ runs and $(n-1)/2$ assents. We know that some algorithm such as sorting are very sensitive to numbers of leaves, runs or assents contained in input data. Therefore, it is necessary to generate random permutations controlling these indexes to conduct precise evaluation of algorithms. This paper describes some methods to generate random permutations with simple indexes which are defined in the paper. The indexes include leaves, runs, assents and some other characteristics of permutations. We give $O(nm)$ algorithms for generating permutations with length n and index m. We also show an $O(n)$ approximate algorithm for generating permutations with length n and leaves m.

1 はじめに

整列 (ソーティング) やデータ検索アルゴリズム等の実用的アルゴリズムの精密な性能評価 [2, 3, 5] 及び高性能確率的アルゴリズムの作成 [9] 等においては，一様性の保証された各種乱順列の生成が必須である．本稿ではそのような乱順列の高速生成法について議論する．ただし，乱順列とは，指定された順列の集合から一様に (即ち，等確率で) 取り出された順列である．また，本稿で扱う順列の性質は，特性指標により表される．そして，順列 s の特性指標とは，s から非負の整数上への関数または関数値である．

順列の長さ自身は自明な特性指標である．また，順列に含まれる逆順 (順列において左側の要素が右側の要素よりも大きい対) の個数，葉 (順列において自分よりも小さい隣人を持たない要素．このような要素を葉と呼ぶ理由については 5 節で明らかにする) の個数，ランズ [11]，上昇部分 (順列において，左側の要素が右側の要素より小さい 2 つの隣接する要素) の個数（即ち n–ランズ）等も特性指標の例である．例えば，順列が 1, 7, 2, 6, 5, 3, 4 の時，長さは 7，逆順の個数は 10，葉数は 3(葉は 1，2，3 である)，ランズは 4 そして上昇部分数は 3 である．

長さ n の順列を一様に (即ち等確率 $1/n!$で) 生成する $O(n)$ アルゴリズムは知られている [11]．しかし，ここで生成される乱順列は非常に偏った性質を有する．例えばその葉数は殆どの場合約 $(n+1)/3$ である [15]．またランズの平均はその性質より明らかに約 $(n+1)/2$ である．従って，例えば一様乱順列を用いて整列法の評価を行うと，QUICKSORT[10] の様に葉数が多い場合に速く，少ない場合に遅い方法に有利になる．一方，MERGESORT[11] や LOAS[7] の様に葉数が少ない場合に速く，多い場合に遅い方法にとって一様乱順列による評価は不利である [7, 5]．実際に整列が行われる問題領域に適合した葉数を持った順列を用いて評価をしないと，実用的評価とは言い難い．現実の世界において整列の対象となる大規模データの葉数を我々は計測していないが，それは [11] にある通り，整列済みに近いと予想される．整列済みに近いデータの葉数は 1 に近く，QUICKSORT が 1 番不得意とする領域である．従って，アルゴリズムの性能を精密に評価するためには，アルゴリズムが実際に扱うデータ領域が持つ特性指標を制御しながら乱順列を生成する必要がある．

特殊な特性指標を持つ乱順列を生成する試みはいくつか報告されているが [2, 3]，それ等は一様性の保証をしていない．例えば [3] では，最長上昇部分列 (LUSS と略記する) の長さを指定して乱順列を生成している．しかしそこでの方法は，複数の LUSS を持つ順列の出現確率の方が，LUSS を 1 つしか持たない順列よりも出現確率が高い．例えば長さ 6 かつ LUSS 長を 3 とした場合，順列 4, 1, 5, 2, 6, 3 は，順列 6, 5, 4, 1, 2, 3 の 4 倍以上の確率で生成される．また [2] では葉数を指定し乱順列を生成しているが，本稿の 7 節で示す通り，χ^2検定での結果が思わしくない．また 2 分木を等確率で生成する方法が [14] で報告されているが，これは特性指標を制御した順列の生成には直接応用できない．これまでの段階においては，指定された葉数を持つ順列を一様に生成した我々の報告 [5] は，自明でない特性指標に関する数少ない実施例の一つと考えられる．

本稿では，[5] での方法を拡張し特定の特性指標ではなく，単純指標と呼ばれるクラスに属する指標を持つ乱順列を生成する方法について議論する．それは直感的には，順列がその生成規則に従って 1 つ短い順列から生成される際に，その値が高々1しか増加しない特性指標である．順列の長さ，葉数，ランズ，上昇部分数等は単純指標の例である．本稿では先ず，順列の生成法と単純指標を形式的に定義する．次にその生成法をランダマイズし，乱順列を生成する方法を提案する．その後，乱順列を生成する際の計算機オーバーフローの問題点について議論する．さらに問題点の解決策として長さ n，単純指標 m の乱順列をオーバーフローを抑えて $O(nm)$ で生成する方法を提案する．最後に，長さ n，葉数 m の乱順列を $O(n)$ で生成する為の実用的な近似方式を提案する．

2 順列の単純指標

ここではまず本稿で扱う順列の生成法を規定する．

特性指標を i，そしてその定義域と値域を各々D_i および V_iとする．ただし，D_iは順列の集合そして V_i

は非負の整数の集合である．また要素 $1, 2, \ldots, n$ の任意の順列を $p = p_1p_2\ldots p_n$ で表す．本稿で扱う順列 s はある順列 p に対し，p_1から出発し，$p_2, p_3, \cdots, p_n$ と順次要素をその時の長さ jの順列とその特性指標 $i(s_j)$ によって決まる場所のうちの一個所に挿入することにより作成出来るものに限る．この順列 p を生成規則の随伴順列と呼ぶ．即ち，特性指標 i と随伴順列 p により生成規則が定まる順列のみを本稿では扱う．

例 1:　与えられた順列の純上昇部分列とは次の条件を満たす上昇部分列である: 部分列中のどの要素も，自分より大きい要素を，自分より左側にある順列の部分に含まない．例えば順列 2, 6, 5, 7, 3, 4 において，先頭から始まる純上昇部分列は次の 3 つだけである：(1) 2 と (2) 2, 6 および (3) 2, 6, 7. 例えば 2, 5, 7 は純上昇部分列ではない．何故ならば，順列中の 5 の左にそれより大きな数 6 があるからである．

長さ n，先頭から始まる最長純上昇部分列 (LPUSS と略記する) の長さ m の順列を生成する場合には，要素を $n, n-1, \ldots, 1$ の順に追加する．長さ n，先頭 LPUSS 長 m の順列は次の 2 つの順列の一方に要素 1 を挿入することによって生成できる:

(1) 長さ $n-1$ かつ先頭 LPUSS 長 $m-1$ の順列の先頭 (1 個所) に要素 1 を挿入する．このとき，長さと特性指標は 1 づつ増加する．

(2) 長さ $n-1$ かつ先頭 LPUSS 長 m の順列の，任意の要素の右側 ($n-1$ 個所) に要素 1 を挿入する．このとき，長さは 1 増加するが特性指標は変化しない．

要素 1 の挿入個所はこの他には存在しないことに注意されたい．長さ $n-1$ で先頭 LPUSS 長 $m-1$ の順列および先頭 LPUSS 長 m の順列は上と同様にして再帰的に作成すればよい．ちなみに，この生成規則から明らかな通り長さ n，先頭 LPUSS 長 m の順列の総数を $S_1(n,m)$ とすると，それは次の再帰方程式で表される:

(1) $S_1(n,m) = 0$
　if $m < 0$, $n < m$ or ($0 < n$ and $m = 0$).

(2) $S_1(0,0) = 1$.

(3) $S_1(n,m) = (n-1)S_1(n-1,m)+S_1(n-1,m-1)$.

ちなみに，一様乱順列の先頭 LPUSS 長の平均は H_n(第 n 番目の調和数) である [15]．また，$S_1(n,m)$ は第 1 種のスターリング数[8] と同じ値を持つ．従って $\sum_{m=1}^{n} S_1(n,m) = n!$ であることに注意されたい [8]．即ち，先頭 LPUSS 長 m は順列全体上で定義された特性指標である．単純指標が順列全体上で定義されるための一般的な必要十分条件は，後述の定理 2 で与えられる．

直感的には上例のように，単純指標とは，順列がその生成規則に従って 1 つ短い順列から生成される際に高々1 しか増加せず，しかも要素の追加可能個所数が特性指標と順列の長さから決定出来るような指標である．その形式的定義は次の通りである．

<u>定義 1</u>　i は，長さ 0 の順列に対する特性指標値が 0 の特性指標とする．このとき，長さ n かつ特性指標 $i(s) = m$ の任意の順列 s を生成する次のような生成規則が存在する時に限り，特性指標 i を単純指標と呼ぶ：ある随伴順列 p が存在し，$s_1 = p_1$ から出発して，$p_j(2 \leq j \leq n)$ を追加することにより $s_2, \ldots, s_n = s$ の順で s が生成できる．ただし次の 3 条件が成立するものとする．

(1) 任意の j ($2 \leq j \leq n$) に対して
　$0 \leq i(s_j) - i(s_{j-1}) \leq 1$.

(2) p_jの挿入可能個所の候補者数が長さ jと特性指標 $i(s_{j-1})$ により一意的に決定できる．

(3) 長さ $n-1$ かつ特性指標 m の順列が作られた時，これに要素 p_nを追加して長さ n かつ特性指標 m の順列を必ず作ることが出来る．

定義 1(3) は，順列を生成する際に，途中で指標が目的の値に到達してしまっても，長さが n になるまではその後の要素の挿入の可能性を保証するものである．

<u>定理 1</u>　長さ n かつ単純指標 m の順列の総数を $S(n,m)$ とする．この時 $S(n,m)$ の定義域の再帰部分は $\{(n,m) \mid 0 \leq m \leq n\}$ の部分集合上で定義され，下記の様な式になる: $S(n,m) = x(n,m)S(n-1,m)+y(n,m)S(n-1,m-1)$．ただし，$x(n,m)$ と $y(n,m)$ は Sへの再帰呼出しを含まず，再帰部分の定義域においては n 以下の非負の整数値をとる関数である．また $S(n-1,m) > 0$ ならば $x(n,m) > 0$ である．(証明略)

上記の $S(n,m)$ を単純指標の特性方程式と呼ぶ．長さ 0 の順列は 1 つしかないと考えれば，単純指標

の条件より $S(0,0)=1$ である．長さ n の順列が持ちうる最大の単純指標を m_0 とすれば，$m_0<m$ なる m に対して $S(n,m)=0$ である．また，$m<0$ または $n<m$ ならば $S(n,m)=0$ である．一方，定理 1 より $S(n,m_0)>0$ ならば $x(n+1,m_0)>0$ であるので，$S(n+1,m_0)>0$ である．従って，長さ $n+1$ の順列が持ちうる最大の単純指標は m_0 以上であることに注意されたい．

例 2:　順列 s に対して $i(s)$ は s に含まれる上昇部分 (左側の要素が右側の要素より小さい 2 つの隣接する要素) 数とする．順列の生成を，$1,2,\ldots,n$ の順で行う．要素 n を既存の長さ $n-1$，上昇部分数 m の順列の上昇部分 a_ja_{j+1} の間又は順列の左端 ($m+1$ 個所) に挿入すれば上昇部分数は増加しない．一方，長さ $n-1$，上昇部分数 $m-1$ の順列の下降部分 b_jb_{j+1} の間又は順列の右端 ($n-m$ 個所) に挿入すれば上昇部分数は 1 増加する．順列における挿入個所はその他にはない．そして $i(s)=m$ なる長さ n の順列 s の総数 $E(n,m)$ は下記の通り定義できる:

(1) $E(n,m)=0$ if $m<0$, $n-1<m$ or $n<0$.
(2) $E(0,0)=1$.
(3) $E(n,m)=(m+1)E(n-1,m)$
$\quad + (n-m)E(n-1,m-1)$.

この時，$x(n,m)=m+1$，$y(n,m)=n-m$．ちなみに，この $E(n,m)$ はオイラー数 (Eulerian numbers) と呼ばれている [8]．$\sum_{m=1}^{n-1}E(n,m)=n!$ であるので，この指標も順列全体に対して定義された単純指標である．

例 3:　順列 s に対して $i(s)$ は s に含まれるランズ (左側の要素が右側の要素より大きい 2 つの隣接する要素数+1) とする．順列の生成を，$n, n-1,\ldots,1$ の順で行う．要素 1 を既存の長さ $n-1$，ランズ m の順列の下降部分 a_ja_{j+1} の間又は順列の左端 (m 個所) に挿入すればランズは増加しない．一方，長さ $n-1$，ランズ $m-1$ の順列の上昇部分 b_jb_{j+1} の間又は順列の右端 ($n-m+1$ 個所) に挿入すればランズは 1 増加する．順列における挿入個所はその他にはない．そして $i(s)=m$ なる長さ n の順列 s の総数 $R(n,m)$ は下記の通り定義できる:

(1) $R(n,m)=0$
if $m<0$, $n<m$, $n<0$ or ($0<n$ and $m=0$).
(2) $R(0,0)=1$.
(3) $R(n,m)=mR(n-1,m)$
$\quad + (n-m+1)R(n-1,m-1)$.

この時，$x(n,m)=m$，$y(n,m)=n-m+1$．定義から当然 $R(n,m)=E(n,m-1)$ である．従って，これも順列全体に対して定義された単純指標である．

次に単純指標とならない特性指標を 2 つ示す．

例 4:　特性指標として，先頭から始まる最長上昇列の長さ m を考える．例えば，順列が $\underline{1}, \underline{2}, \underline{7}, 6, 5, 3, 4$ の時，$m=3$(下線の部分)，順列が $1, 2, 3, 4, 5, 6, 7$ の時，$m=7$，そして順列が $7, 1, 2, 3, 4, 5, 6$ の時，$m=1$ である．この m は単純指標ではない．何故ならば，どんな随伴順列を考えても，新たな要素を先頭上昇列に追加することにより，その長さ即ち特性指標を減少させる可能性があるからである．これは単純指標の条件 (1) に反する．

例 5:　順列 s に対して $i(s)$ は s に含まれる最長上昇部分列長とする．任意の随伴順列 p に対し，長さ n かつ最長上昇部分列長 m の順列の生成を $p_1,\ldots,p_n$ の順で行う．要素 p_n を挿入するとき，既存の最長上昇部分列の右端の要素のうちで一番左側にあるものを見付けそれを a とする．要素 p_n が a より大きい時，a より左側に挿入すれば，最長上昇部分列長は変わらない．a より右に挿入すれば最長上昇部分列長は 1 増加する．ところが，a の位置は順列の長さと特性指標だけからは決まらないことがある．例えば，長さ 7 かつ最長上昇部分列長 3 の 2 つの順列 $7, 6, 5, 4, 1, 2, 3$ と $5, 6, 7, 4, 3, 2, 1$ に 8 を挿入する場合を考える．最長上昇部分列長を 1 増やす挿入の仕方は，前者では 1 通り (3 の右) しかないが，後者では 5 通り (7 より右なら何処でも良い) ある．いかなる順列についても同様なことが起こるので，これは単純指標の条件 (2) に反する．p_n が a より小さい場合も同様である．従って最長上昇部分列長は単純指標ではない．

順列に含まれる葉数は単純指標である．この節の終わりとして，単純指標の定義域に関する次の定理を与える．

<u>定理 2</u>　単純指標 i の定義域が順列全体であるための必要十分条件は下記の 2 条件である: (1) 特性方程式の係数の間に関係 $y(n,m)=n-x(n,m-1)$ があり，しかも (2) 長さ n の順列が持ちうる最大の指

標 m_0 に対して $S(n-1, m_0) = 0$ または $x(n, m_0) = n$. (証明略)

例 1 から 3 までで述べた特性指標および後述の葉数の特性方程式は全て定理 2 の条件に当てはまることに注意されたい．長さ n かつ特性指標 m の順列全体に番号を振る方法は文献 [5] の方法を一般化したものを用いる．これは，乱順列生成法を χ^2 検定する際に利用できる (8 節表 1 参照).

3 乱順列生成上の問題点

指定された単純指標 m を持つ長さ n の順列をランダム，即ち等確率で生成する場合について考える．随伴順列を p かつ特性方程式を定理 1 の $S(n,m)$ とする．単純指標の定義より $p_1, \cdots, p_n$ をその順で挿入することにより長さ n かつ単純指標 m の任意の順列を生成する事が出来る．各順列を等確率で生成するためには下記の様な再帰計算を行えば良い:

(1) 一様乱数に基づいて，下記 (2) または (3) のどちらかをその出現確率に応じて選ぶ.

(2) 確率 $x(n,m)S(n-1,m)/S(n,m)$ で要素 p_n を，長さ $n-1$ かつ単純指標 m の乱順列に指標を増加させないように挿入する．この長さ $n-1$ かつ単純指標 m の乱順列は再帰的に生成する．挿入個所は $x(n,m)$ あるので，その内の 1 個所をランダムに選ぶ.

(3) 確率 $y(n,m)S(n-1,m-1)/S(n,m)$ で p_n を，長さ $n-1$ かつ単純指標 $m-1$ の乱順列に指標を増加させるように挿入する．この乱順列は再帰的に生成する．挿入個所は $y(n,m)$ あるので，その内の 1 個所をランダムに選ぶ.

この計算では，出現確率に応じて生成されたランダムな部分順列に，新しい要素をランダムに挿入しているので，生成される順列の出現確率は $1/S(n,m)$，即ち，等確率である．ここでの問題は，確率 $x(n,m)S(n-1,m)/S(n,m)$ の計算法である．何故ならば，一般に関数 S の値は大きな n に対しては非常に大きくなる．例えば後述の関数 $L(n,m)$ の値は $\Omega(2^n)$ である．従って，分母と分子を別々に計算した後に割り算をすることは計算機オーバーフローにより不可能である．

一方，確率の値が計算機アンダーフローを起こすほど小さくなる場合は少ない．例えば $2mL(n-1,m)/L(n,m) \geq (n-2m+1)/(n-m)$ であり (定理 10)，計算機で楽に扱える範囲の値である．したがって，確率を分子と分母に分けずに直接計算する必要がある．以下では，確率の逆数 $S(n,m)/\{x(n,m)S(n-1,m)\}$ を構成比と呼び，確率の代わりにこれを扱う．その理由は，構成比は 1 以上であり，値が 1 以下である確率よりも扱い易いからである．

4 構成比の直接計算法

この節では構成比，即ち出現確率の逆数，の直接計算法について議論する．本節で扱う関数 $S(n,m)$ は単純指標の特性方程式である．それは整数の対 (n,m) 上で定義された 0 以上の値を持つ整数値関数であり，その定義式は次の 3 部分からなるものとする．

(1) S の値として 0 を返す非再帰部分: その終了条件を $C_0(n,m)$ とする．S の値は $0 \leq m \leq n$ 以外では必ず 0 であるから，$\neg(0 \leq m \leq n)$ ならば $C_0(n,m)$ である．また S は C_0 を満たす n，m 以外では 0 にならないものとする．即ち $C_0(n,m)$ の時に限り $S(n,m) = 0$.

(2) S の値として非 0 を返す非再帰部分: その終了条件を $C_1(n,m)$ とする．即ち $C_1(n,m)$ ならば $S(n,m) > 0$.

(3) S の再帰部分: $S(n,m) = x(n,m)S(n-1,m) + y(n,m)S(n-1,m-1)$ とする．関数 $x(n,m)$ と $y(n,m)$ は再帰部分の定義域において 0 以上の値を取る関数であり，両者とも S への再帰呼出しを含まないものとする．ただし，定義 1(3) で述べた通り $S(n-1,m) > 0$ ならば $x(n,m) > 0$ である．ここで S が 0 になる点を特別扱いしているのは，構成比の分母が 0 になる点を除く為である．

前述の例 1, 例 2 および例 3 における 3 つの関数，S_1，E および R は上記の条件を満たす．以下にその他の例を 2 つあげる (これ等の関数を特性方程式として持つ単純指標については，5 節の例 11，12 で議論する)．上記 S に関する 3 つの各場合に対応させて，(1)〜(3) の 3 つの場合に分けて各例は示し

である.

例 6: 組合せ関数 (Combination) ${}_nC_m = C(n,m)$

(1) $C(n,m) = 0$ if $m < 0$ or $n < m$.

(2) $C(0,0) = 1$.

(3) $C(n,m) = C(n-1,m) + C(n-1,m-1)$
この時, $x(n,m) = y(n,m) = 1$.

例 7: 第 2 種スターリング数 (Stirling numbers of the second kind) $S_2(n,m)$

(1) $S_2(n,m) = 0$ if $n < m$, $m < 0$ or ($m = 0$ and $0 < n$).

(2) $S_2(0,0) = 1$.

(3) $S_2(n,m) = mS_2(n-1,m) + S_2(n-1,m-1)$
この時, $x(n,m) = m$, $y(n,m) = 1$.

<u>定理 3</u>　上記の再帰関数 S に対しその構成比 F を次のように定義する: $F(n,m) = S(n,m)/\{x(n,m)S(n-1,m)\}$. この時, Fは $S(n-1,m)$ が 0 でない領域, 即ち$\neg C_0(n-1,m)$ なる n, m において, (1) 非再帰部分, (2)Fへの再帰呼出しが一回しか現れない線形再帰部分, (3)Fへの再帰呼出しが 2 回現れる木再帰部分および (4)Sへの呼出しが残る部分の 4 部分により再帰的に定義できる. ただし以下では, $CoefF(n,m) = x(n-1,m-1)y(n,m)/(x(n,m)y(n-1,m))$ とおく (証明略):

(1) $C_0(n-1,m-1)$ なる n, m に対し $F(n,m) = 1$ で停止する非再帰部分.

(2) $\neg C_0(n-1,m-1) \cap C_0(n-2,m)$ なる n, m に対応する線形再帰部分: $F(n,m) = 1 + CoefF(n,m)F(n-1,m-1)$.

(3) $\neg C_0(n-1,m-1) \cap \neg C_0(n-2,m) \cap \neg C_0(n-2,m-1)$ なる n, m に対応する木再帰部分 : $F(n,m) = 1 + CoefF(n,m)F(n-1,m-1)(F(n-1,m)-1)/F(n-1,m)$

(4) $\neg C_0(n-1,m-1) \cap \neg C_0(n-2,m) \cap C_0(n-2,m-1)$ なる n, m に対応する部分, 即ち $F(n,m) = 1 + y(n,m)y(n-1,m-1)S(n-2,m-2)/\{x(n-1,m)S(n-2,m)\}$

定理 3 において (4) 以外の場合には F の定義中に Sは現れない. 従って (4) の場合が存在しないか, あるいは存在する場合でも Sの固有の性質を利用して Sを消去することが出来れば, S無しで F を定義することが可能となる. なお, (2)~(4) で $\neg C_0(n-1,m-1)$ の全ての場合をカバーしているので, (1)~(4) では$\neg C_0(n-1,m)$ の全ての場合をカバー出来ていることに注意されたい.

<u>定理 4</u>　定理 3 により生成される関数 $F(n,m)$ において, 場合 (4) が空ならば $F(n,m)$ は時間・スペース共に $O(mn)$ で計算できる. (証明略)

定理 3 の適用例 8~10 を以下に示す.

例 8: 組合せ関数の構成比

$F_c(n,m) = C(n,m)/C(n-1,m)$ の定義域は $0 \le m < n$ であり, 定理 3 に対応する各場合は下記の通りである.

(1) $F_c(n,0) = 1$.

(2) $F_c(n,n-1) = 1 + F_c(n-1,n-2)$
(1) を利用してこの線形再帰方程式を解けば $F_c(n,n-1) = n$ が得られる.

(3) $F_c(n,m) = 1 + F_c(n-1,m-1)(F_c(n-1,m)-1)/F_c(n-1,m)$.

(4) $C(n-2,m) > 0$ and $C(n-1,m-1) > 0$ ならば $m-1 \ge 0$ かつ $n-2 \ge m$. 従って $m-1 \ge 0$ かつ $n-2 \ge m-1$ より $C(n-2,m-1) > 0$. 従って, 場合 (4) は存在しない.

従って $0 \le m < n$ の時 $F_c(n,m)$ の直接計算法は下記の通りとなる:

(1) $F_c(n,0) = 1$.

(2) $F_c(n,n-1) = n$.

(3) $F_c(n,m) = 1 + F_c(n-1,m-1)(F_c(n-1,m)-1)/F_c(n-1,m)$.

(注意: 実は $F_c(n,m) = n/(n-m)$ であるが, 上記の再帰方程式よりこの式を求めるプログラム変換法を我々はまだ開発していない).

例 9: 第 2 種スターリング数の構成比

$F_s(n,m) = S_2(n,m)/mS_2(n-1,m)$ の定義域は $0 < m < n$ であり, 定理 3 に対応する各場合は下記の通りである.

(1) $F_s(n,1) = 1$ if $n > 1$.

(2) $F_s(m+1,m) = 1 + ((m-1)/m)F_s(m,m-1)$ for $m > 1$.
ここで $Fs(m+1,m)$ が下記の関数 $g(m)$ と等価であることに注意されたい.
$g(2) = 3/2$
$g(m) = 1 + ((m-1)/m)g(m-1)$
この線形再帰方程式を解けば $g(m) = F_s(m+1,m) = (m+1)/2$ for $m > 0$ が得られる.

(3) $F_s(n,m) = 1 + ((m-1)/m)F_s(n-1,m-1)(F_s(n-1,m)-1)/F_s(n-1,m)$

(4) $S_2(n-2,m) > 0$ and $S_2(n-1,m-1) > 0$ ならば $n-2 \geq m > 1$．従って，$S_2(n-2,m-1) > 0$ であり，場合 (4) は存在しない．

従って $0 < m < n$ の時 $F_s(n,m)$ の直接計算法は下記の通りとなる:

(1) $F_s(n,1) = 1$ if $n > 1$.

(2) $F_s(m+1,m) = (m+1)/2$ for $m > 0$.

(3) $F_s(n,m) = 1 + ((m-1)/m)F_s(n-1,m-1)(F_s(n-1,m)-1)/F_s(n-1,m)$.

このスターリング数の構成比の n を無限大にした時の極限値が 1 であることは知られている [1]．しかし，任意の n と m に関する閉じた式は我々の調査の範囲では見出すことが出来なかった．下記のオイラー数についても同様である．

例 10: オイラー数の構成比

$F_e(n,m) = E(n,m)/((m+1)E(n-1,m))$ の定義域は $0 \leq m < n-1$ であり，定理 3 に対応する各場合は下記の通りである．

(1) $F_e(n,0) = 1$.

(2) $F_e(m+2,m)$

$$= 1 + \{2*m/((m+1)*1)\}F_e(m+1,m-1)$$
$$= 1 + (2m/(m+1))F_e(m+1,m-1) \quad \text{for } m > 0.$$

ここで $F_e(m+2,m)$ が下記の関数 $g(m)$ と等価であることに注意されたい．

$g(1) = 2$

$g(m) = 1 + (2m/(m+1))g(m-1)$

この線形再帰方程式を解けば $g(m) = F_e(m+2,m) = (2^{m+2}-m-3)/(m+1)$ が得られる．

(3) $F_e(n,m)$

$$= 1+\{m*(n-m)/((m+1)*(n-m-1))\}F_e(n-1,m-1)(F_e(n-1,m)-1)/F_e(n-1,m)$$
$$= 1+\{m(n-m)/((m+1)(n-m-1))\}F_e(n-1,m-1)(F_e(n-1,m)-1)/F_e(n-1,m).$$

(4) $E(n-2,m) > 0$ and $E(n-1,m-1) > 0$ ならば $n-2 > m \geq 1$．

従って，$E(n-2,m-1) > 0$ であるので場合 (4) は存在しない．

従って $0 \leq m < n-1$ の時 $F_e(n,m)$ の直接計算法は下記の通りとなる:

(1) $F_e(n,0) = 1$.

(2) $F_e(m+2,m) = (2^{m+2}-m-3)/(m+1)$.

(3) $F_e(n,m) = 1 + \{m(n-m)/((m+1)(n-m-1))\}F_e(n-1,m-1)(F_e(n-1,m)-1)/F_e(n-1,m)$.

定理 3 の構成比 F を実際にコンピュータで計算するとき，場合 (3) の式 $F(n-1,m)-1$ で深刻な桁落ちが発生し，正しい構成比が求まらなくなることがある．構成比の符号が負になることすら起こりうる．そこで桁落ちを防ぐために，F の計算法を次の系 1 のように書き換える．

<u>系 1</u>　$F(n,m) = 1 + F_q(n,m)$ とおく．定理 3 の場合 (4) が空のとき，F_q の値は下記の再帰方程式で計算出来る（証明略）:

(1) $C_0(n-1,m-1)$ なる n，m に対し $F_q(n,m) = 0$.

(2) $\neg C_0(n-1,m-1) \cap C_0(n-2,m)$ なる n，m に対し $F_q(n,m) = CoefF(n,m)(F_q(n-1,m-1)+1)$.

(3) $\neg C_0(n-1,m-1) \cap \neg C_0(n-2,m) \cap \neg C_0(n-2,m-1)$ なる n，m に対し $F_q(n,m) = CoefF(n,m)(F_q(n-1,m-1)+1)F_q(n-1,m)/(F_q(n-1,m)+1)$.

次には，構造帰納法 [12] を用いて構成比 F の単調性を調べる．ただし，F に対応する定理 3(4) の場合は空とする．先ず，F の定義域上での構造帰納法の方法について簡単に説明する．F の定義域上，即ち $\neg C_0(n-1,m)$ の任意の要素 (n,m) についてある性質 $\phi(n,m)$ が成立することを証明するためには次の 3 つの段階により行う．この各段階は，定理 3 の (1)〜(3) に対応する．

(1) $C_0(n-1,m-1)$ なる n，m に対し $\phi(n,m)$ が成立することを証明する．

(2) $\neg C_0(n-1,m-1) \cap C_0(n-2,m)$ なる n，m に対し，$\phi(n-1,m-1)$ の成立を仮定して $\phi(n,m)$ を証明する．

(3) $\neg C_0(n-1,m-1) \cap \neg C_0(n-2,m) \cap \neg C_0(n-2,m-1)$ なる n，m に対し，$\phi(n-1,m-1)$ および $\phi(n-1,m)$ を仮定して $\phi(n,m)$ を証明する．

<u>定理 5</u>　定理 3 で扱った再帰関数 S と構成比 F について考える．定理 3 の場合 (1) のとき即ち，$F(w,v) = 1$ ならば $F(w+1,v) = 1$ かつ定理 3

の場合 (4) が空かつ関数 Fの再帰部分の定義域に属する任意の (n,m) に対して $CoefF(n+1,m) \leq CoefF(n,m)$ ならば，Fの定義域上の任意の (n,m) に対して $F(n+1,m) \leq F(n,m)$.（証明略）

<u>定理 6</u>　定理 3 で扱った再帰関数 Sと構成比 F について考える．ただし，$\neg C_0(n-1,m) \cap C_0(n-2,m)$ ならば $C_0(n-2,m+1)$ とする．このとき定理 3 の場合 (4) が空かつ関数 Fの再帰部分の定義域に属する任意の (n,m) に対して $CoefF(n,m) \leq CoefF(n+1,m+1)$ ならば，F の定義域上の任意の (n,m) に対して $F(n,m) \leq F(n+1,m+1)$.（証明略）

<u>定理 7</u>　定理 3 で扱った再帰関数 Sと構成比 F について考える．ただし，$\neg C_0(n-1,m) \cap C_0(n-2,m)$ ならば $C_0(n-2,m+1)$ とする．このとき定理 3 の場合 (4) が空かつ関数 Fの再帰部分の定義域に属する任意の (n,m) に対して $CoefF(n,m) \leq CoefF(n,m+1)$ ならば，Fの定義域上の任意の (n,m) に対して $F(n,m) \leq F(n,m+1)$.（証明略）

<u>系 2</u>　第 2 種スターリング数の構成比 F_sは，n に関して単調減少かつ m に関して単調増加である．（証明略）

<u>系 3</u>　オイラー数の構成比 F_eは，n に関して単調減少かつ m に関して単調増加である．（証明略）

5　順列の葉数

葉数については文献 [5] で詳説した．ここでは，葉数 (m) と上昇部分数 (k) との関係について触れる．2 次節 $(m-1$ 個$)$ があれば必ず上昇部分がありかつ，上昇部分の個数は 2 次節と 1 次節の個数 $(n-2m+1)$ を合わせたもの (即ち $n-m$) 以下であるから下記の定理が成り立つ．

<u>定理 8</u>　n, m および kの間には関係 $m-1 \leq k \leq n-m$ が成立する．

オイラー数 $\mathrm{E}(n,k)$ の再帰方程式の主要部は $E(n,k) = (k+1)E(n-1,k) + (n-k)E(n-1,k-1)$ であり，$L(n,m)$ と良く似ている．それにもかかわらず，我々は Eと L の直接的関係については分かっていない．しかし両者とも下記の関数 $A(n,m,r)$ の級数として表現できることを発見した．ただし rは右側が空いている (即ち上昇部分になっている)1 次節の個数である (この時上昇部分数 $k=r+m-1$ に注意).

<u>定義 3</u>　長さ n，葉数 m の順列のうちで右側が空いている 1 次節を r個持つものの総数を $A(n,m,r)$ とすれば，A は下記のように定義できる:

(1) $A(n,m,r) = 0$ if $n < 2m-1$, $m < 1$, $r < 0$ or $r > n-2m+1$.

(2) $A(n,1,0) = 1$.

(3) $$\begin{aligned} A(n,m,r) = {} & mA(n-1,m,r-1) \\ & + mA(n-1,m,r) \\ & + (r+1)A(n-1,m-1,r+1) \\ & + (n-2m+2-r)A(n-1,m-1,r). \end{aligned}$$

この方程式の妥当性は次のように説明できる．(1) は明らかであろう．(2) は逆順に並んだ順列を表す．(3) の説明は次の通りである．数列に含まれる上昇部分数は，数列の 2 次節と，右側のあいた 1 次節の個数の和に等しい．従って長さ n，葉数 m，右側が空いている 1 次節の個数 rの対称ヒープは，長さ $n-1$ の対称ヒープに次の 4 通りの方法で 1 を挿入することにより作られる: (1) 葉数 m の対称ヒープの葉の左側に挿入される．この場合葉数は変わらないが，rが 1 増加する．(2) 葉数 m の対称ヒープの葉の右側に挿入される．この場合は葉数も rも変化しない．(3) 右側が空いている 1 次節に挿入される．この場合葉数は増加するが，rは減少する．(4) 左側が空いている 1 次節に挿入される．この場合葉数は増加するが，rは変わらない．これ等の場合は上記定義における式 (3) の，右辺の 4 項にその順番で対応する．新しい要素の挿入方法は上記 4 通り以外に存在しないから，式 (3) は正しい．

□

<u>定理 9</u>　関数 A と組み合わせ，L および Eとの関係は下記の通りである（証明略）:

(1) $A(n,1,r) = {}_nC_r$.

(2) $L(n,m) = \sum_{r=0}^{n-2m+1} A(n,m,r)$.

(3) $E(n,k) = \sum_{m=1}^{\min\{n-k,k+1\}} A(n,m,k-m+1)$.

この節の最後として，一部の順列のみを定義域とする単純指標の例を 2 つ示す．

例 11:　C順列と呼ぶ次のような特殊な順列のクラスを再帰的に定義する．(1)n は C順列である．(2) 要素 n, $n-1, \ldots, 2$ で作られた C順列の左端の葉の左隣または右端の 1 次節に 1 を挿入して作られた順列は C順列である．この時，長さ n かつ葉数 m の C順列は，(1) 長さ $n-1$ かつ葉数 m の C

順列の左端の葉の左側に 1 を挿入したものか，または (2) 長さ $n-1$ かつ葉数 $m-1$ の C順列の右端の 1 次節に 1 を挿入して作られる．従ってその特性方程式は例 6 の組み合わせ関数の再帰部分と一致する．従ってこの時の葉数は C順列を定義域とし，$n, n-1, \ldots, 1$ を随伴順列とする単純指標である．C順列の特性方程式の非再帰部分は組合せ関数のものと一致しないので，方程式の値も一致しないことに注意されたい．例えば，任意の n に対して葉数 1 の C順列の個数は 1 である．また，長さ 5，葉数 2 の C順列は 2, 3, 4, 5, 1 と 1, 3, 4, 5, 2 および 1, 2, 4, 5, 3 の 3 つのみである．5 が葉であるから 4 は 5 の左側にしか付くことができないことに注意されたい．

例 12:　S_2順列と呼ぶ次のような特殊な順列のクラスを再帰的に定義する．

(1) n は S_2順列である．

(2) 要素 $n, n-1, \ldots, 2$ で作られた S_2順列の任意の葉の左隣または右端の 1 次節に 1 を挿入して作られた順列は S_2順列である．

この時，長さ n かつ葉数 m の S_2順列は，(1) 長さ $n-1$ かつ葉数 m の S_2順列の葉の左側に 1 を挿入したものか，または (2) 長さ $n-1$ かつ葉数 $m-1$ の S_2順列の右端の 1 次節に 1 を挿入して作られる．従ってその特性方程式は例 7 の第 2 種スターリング数の再帰部分と一致する．従ってこの時の葉数は S_2順列を定義域とし，$n, n-1, \ldots, 1$ を随伴順列とする単純指標である．C順列と同様，長さ n かつ葉数 1 の S_2順列の個数は 1 であることに注意されたい．

6　長さ n 葉数 m の順列の構成比計算法

長さ n，葉数 m の $L(n,m)$ 個の順列を等確率で生成する方法は文献 [5] で述べた．

この節では定理 3 を応用し，確率関数 Pの逆数として関数 L の構成比 F_lを用いる事を考える．$P(n,m)$ は 1 が t'の葉に付いている確率 $2m * L(n-1,m)/L(n,m)$ である．ただし，集合 $\{1, 2, 3, \ldots, n\}$ の順列 t の葉数が m であり，かつ t は集合 $\{2, 3, \ldots, n\}$ の順列 t'に 1 を挿入して作られているものとする．

$P(n,m)$ の直接計算法は下記の通りである．

例 13: 関数 L の構成比 F_l

$F_l(n,m) = L(n,m)/(2mL(n-1,m))$ かつ $C_0(n-1,m) \equiv (m < 1 \text{ or } n < 2m)$ として，定理 3 を L に適用すれば下記の通り F_lの再帰的定義が得られる．以下では $CoefF_l(n,m) = (m-1)(u+1)/(mu)$ とおく (ただし $u = n-2m+1$):

(1) $F_l(n,1) = 1$.

(2) $F_l(2m,m) = 1+2\{(m-1)/m\}F_l(2m-1,m-1)$.

(3) $F_l(n,m) = 1 + CoefF_l(n,m)F_l(n-1,m-1)(F_l(n-1,m)-1)/F_l(n-1,m)$.

(4) $L(n-2,m) > 0$ and $L(n-1,m-1) > 0$ ならば $m \geq 2$ かつ $n \geq 2m+1$．従って，$L(n-2,m-1) > 0$ より場合 (4) は存在しない．

従って $1 \leq m$ かつ $2m \leq n$ の時

(1) $F_l(n,1) = 1$.

(2) $F_l(2m,m) = 1+2\{(m-1)/m\}F_l(2m-1,m-1)$.

(3) $F_l(n,m) = 1 + CoefF_l(n,m)F_l(n-1,m-1)(Fl(n-1,m)-1)/Fl(n-1,m)$.

ちなみに F_lの値の大きさは，例えば $F_l(1998, 99) = 1.000006$，$F_l(4001, 2000) = 811.0832$．ただし F_lの値が 1 に近づくと F_lの定義式の (3) における $F_l(n-1,m)-1$ の計算で深刻な桁落ちが発生し，正しい値を求めることができなくなる．さらに都合の悪いことにこの値が負になり，結果として求まる Pの値が 1 を越える可能性が生じる．従って前述の系 1 により，F_lの計算は下記の補助関数 F_p を用いて行う必要がある．

<u>系 4</u>　$F_l(n,m) = F_p(n,m) + 1$，ただし以下では $CoefF_p(n,m) = (m-1)(u+1)/(mu)$ とおく (ただし $u = n-2m+1$):

(1) $F_p(n,1) = 0$.

(2) $F_p(2m,m) = 2\{(m-1)/m\}(F_p(2m-1,m-1)+1)$.

(3) $F_p(n,m) = CoefF_p(n,m)(F_p(n-1,m-1)+1)F_p(n-1,m)/(F_p(n-1,m)+1)$.

$1 \leq m$ かつ $2m \leq n$ であるので，$F_p(n,m)$ の計算をパスカルの 3 角形を用いて行うために必要なスペースは，底辺長 $n-m$，上辺長 $n-2m+1$ そして高さ $m-1$ の台形である．従って所要時間および

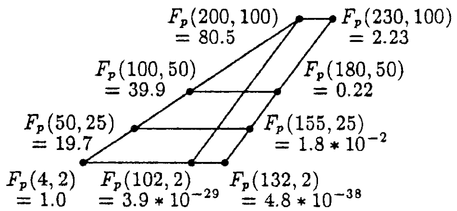

図 1: Fpの値の例
縦に m を，そして横に n を取り，幾つかの引き数 n，m に対する $F_p(n,m)$ とその値を示した．例えば $F_p(200,100) = 80.5$ である．ちなみに $F_p(n,2) = (n-2)/(2^{n-1}-2n+2)$.

スペース所要量は共に $(m-1)(2n-3m)/2$ である(図1参照).

以上で述べた構成比計算法によれば，先ず $O(nm)$ 時間およびスペースで必要な確率関数 Pの値の表を作成し，その表に基づいて乱順列を 1 つあたり $O(n)$ で生成する事が出来る．しかし nが大きな値(例えば 10 万)をとる時，この方法では少なくともメモリネックになる．この場合には，次に述べる，近似計算が必要である．

7 構成比の近似計算

この節では構成比の近似値を $O(1)$ 時間およびスペースで計算する方法について議論する．この方法は，前節で求めた関数 F_pの上界および単調性に基づく．次の定理 10 は $F_p(n,m)$ の有益な上界を与える．以下では $u = n-2m+1$ とする．

定理 10　$n \geq 2m$ ならば $F_p(n,m) \leq (m-1)/u$. (証明略)

定理 11　関数 F_pは n に関して単調減少かつ m に関して単調増加である．(証明略)

関数 F_pの下限は当然0である．さらに関数値を実際に計算してみると次の 2 性質を持つことが分かる(それ等の性質に関する数学的な証明は得られていない).

(1) $F_p(2m,m)$ の値は m が 10 以上では約 $0.8m$ である (mが大きくなるほど $0.8m$ に近づく).

(2) $F_p(3m-2,m)$ の値は m が 10 以上で約 0.5 である (m が大きくなるほど 0.5 に近づく).

上記の性質を利用し，次の関数 F_2を F_pの近似式として求めた．$2m \leq n < 3m-2$ の時 $F_2(n,m) = 0.75^{(n-m)/m} * (m-1)/u$，$3m-2 \leq n$ の時 $F_2(n,m) = 0.75^{n/m} * (m-1)/u$

F_2の値は，$n = 2m$ では $0.8m$ 以下であり，$n = 3m-2$では約0.5であり，かつ全体としては $(m-1)/u$ 以下である．そして F_pと同じ単調性を有する．この近似計算により新たに挿入すべき要素が葉に付く確率を定数時間で計算できるので，長さ n，葉数 m の乱順列を $O(n)$ 時間およびスペースで生成できる．ただし $P(n,2)$ に対しては，その正しい値 $1-(n-2)/(2^{n-1}-n)$ が分かっているので，近似式ではなくその値を使う．Pによる正しい値に対する相対誤差と絶対誤差を調べその結果を表 2に示した．近似方式の精度の比較対象として，次のような簡便法を考える．これは順列に新たな要素を追加する際，要素を葉に付けるか，1 次節に付けるか決定する時に，順列に付いている葉と 1 次節の個数の比を使うものである．挿入の順序は $n, n-1, \ldots, 1$ である．現在の順列の長さを i，葉数を jとする．j が m より小さい間は $2j/(i+1)$ の確率で葉に挿入し，$(i+1-2j)/(i+1)$ の確率で 1 次節に挿入する．即ちこれは，挿入できる個所にランダムに挿入することと同じである．葉数 jが m になったら，もう葉を増やさないために残りの要素は葉に挿入する．また順列の長さ i が $n-m+j$になったらその後は葉を増やすために 1 次節に挿入する．この簡便法では，確率の計算が定数時間で出来るので，長さ n の順列は $O(n)$ 時間で生成出来る．これは，一見公平そうであるが，良く調べると実は偏った順列生成を行う．例えば，$L(5,2) = 88$ であるから長さ 5, 葉数 2 の順列 3, 5, 4, 2, 1 および 5, 4, 2, 3, 1 は共に 1/88 の確率で生成されなければならない．しかし簡便法では，前者の生成確率は 1/96 であり，後者のそれは 1/72 である．即ち，順列の生成確率が一様ではない．

この簡便法は，先に述べた確率を用いる方法が，葉を付けるべき順列木を再帰的に作るボトムアップ的生成法であったのとは逆に，木の根から要素を付けていくトップダウン的生成法である．トップダウン的生成法においても，正しい確率を計算する方法は前と同様にして求める事が出来る．簡便法はその正しい確率を $2j/(i+1)$ で大雑把に近似したものである．正しい確率とその近似値との相対誤差および絶対誤差を，参考のために表 2に示した．

表 1: 長さ n 葉数 m の順列 k個を生成する場合の4方式の比較

方式	時間	空間	χ^2統計量	葉数
確率関数	$n*\max\{m,k\}$	$n*m$	2861.4	16691
近似方式	$k*n$	n	4941.1	16641
簡便法	$k*n$	n	11178.2	16706
浅野方式	$k*n*\log n$	n	25140.0	16652

表 2: 誤差評価

n	m	相対誤差		絶対誤差	
		近似方式	簡便法	近似方式	簡便法
50	25	0.034	0.966	0.014	0.097
100	50	0.028	1.207	0.012	0.121
230	100	0.023	1.212	0.011	0.121

図1における $(50,25)$, $(100,50)$ および $(230,100)$ より左下の領域に含まれる点 (即ち，それ等の点を求めるために必要な全ての点) における正しい確率に対する近似方式と簡便法の相対誤差および絶対誤差の平均を示した.

8 乱順列の検定

長さの短い順列が一様に生成されたか否かを調べる場合には，χ^2検定が使える．即ち，順列に1から $L(n,m)$ までの番号 (木番号と呼ぶ) を付け，木番号の列についてχ^2検定を行う．同時にまたその木番号列の葉数も調べる (順列に含まれる葉数の平均は $(n+1)/3$)．木番号付けの方法は文献 [5] の方法を一般化すればよい．この方法は長さ n, 葉数 m の順列に対応する順列木に番号を付けるものであり，[11] における階乗進数とは異なることに注意されたい.

確率関数を正確に計算したもの, 近似式を用いたもの，前節の最後に述べた簡便法および先駆的業績である浅野方式 [2] の4方式により，長さ7葉数3の順列 ($L(7,3)=2880$) を 150000 個づつ生成した．木番号付け関数 i_1 により各順列に木番号付けして1と2880の間の乱数列を生成した．すなわち，自由度2879で長さ50000の数列を3本作り，各々に対してχ^2検定を行った．表1はその平均値を示したものである．また3本の数列の葉数も1本当たりの平均を示した (葉数の理論的期待値は約16667)．時間とスペースに関するビッグオーも示した.

表1において，確率関数方式は，予想通り良い検定結果を示している．統計量は2879に，そして葉数は16667に近いほど良い．[13] によれば，数列を乱数列とみなせる限界は自由度 (この場合は2879) から$\pm 2\sqrt{\text{自由度}}$以内に統計量が収まることという目安を与えている.

今の場合この値は約±107であり，簡便法と浅野方式は統計量が大きすぎ，順列を一様に発生させるには不適であることが分かる．それ等に比べ，近似方式の統計量はあまり大きくないので，前2者よりは優れており，適当にランダムな順列を生成する際には利用できると思われる.

もっと大きな順列，例えば $n=4095$, $m=2048$ のとき，生成される乱数列は 1516^{2047}以上の自由度を持ち，その一様性をχ^2検定することは不可能である．また，葉数を指定された順列は特殊であるため，一様乱順列の検定に利用される順列検定法 [11] を利用することはできない．一方，確率関数方式における計算法で求められた $P(n,m)$ の値は，計算機誤差の範囲において正しい．この値と近似方式および簡便法との相対誤差と絶対誤差を調べ，表2に示した.

表2より，簡便法の相対誤差は近似方式と比べてはるかに大きい (約30倍以上) ことが分かる．これからも簡便法が一様乱順列の生成に不適なことが理解できよう．近似方式は n が50以上の場合は数パーセント程度の誤差を含む．表には示されていないが，nが小さい時は，近似方式の誤差はそれより大きい．ちなみに，χ^2検定を行なった $(7,3)$ に対する平均相対誤差は，0.06であった.

近似方式は n が大きいほど精度が良くなるのに対し，簡便法は n が大きくなるほど精度が下がる傾向がある．精度の高い乱順列を生成する際には確率関数を利用する必要がある．しかし高い精度が必要ないとき，メモリ不足の時，あるいは長い乱順列を高速に生成したい時には近似方式を利用すればよいと考えられる.

9 おわりに

特性方程式 $S(n,m) = x(n,m)S(n-1,m) + y(n,m)S(n-1,m-1)$ の構成比 $F(n,m) = S(n,m)/\{x(n,m)S(n-1,m)\}$ を Sの値を計算することなく直接求める $O(nm)$ 時間の方法を示した．またその実用的応用例として，長さ n 葉数 m の順列に対する構成比直接計算法について詳述した．n

が数100迄の順列であるならばその方法を利用し，計算機オーバーフロー無しで高速に乱順列を生成できる．しかし，それ以上に大きな n に対しては構成比の近似式を求めるなどして，計算の高速化を図らないと時間とスペースが掛かりすぎて現実的でない．本稿の方式によれば，構成比が直接的に再帰関数として表現出来るので，その性質(単調性，上界，下界等)を構造帰納法[12]で証明し易い(例えば定理5，6，7)．我々は実際その様にして，関数 F_p の精度の高い近似式を求め，長さ n 葉数 m の乱順列を $O(n)$ で生成し整列法の評価を行なった．これにより，長さ10万以上の大きなデータを用いて整列アルゴリズムの精密な評価が可能となった．

アルゴリズムの更に精密な評価をするためには，葉数とランズのように複数の特性指標を同時に制御して乱順列を生成する必要がある．そのためには，複数の特性指標をパラメタとし，しかも項数のより多い再帰方程式，たとえば本稿第5節定義3における A のような関数の構成比高速計算法の開発が今後の課題である．

謝辞

本研究に伴うプログラム開発に協力し，かつ有益な議論をして下さった早稲田大学大学院理工学研究科遠藤貢一氏に深謝します．

参考文献

[1] Abramowitz, J. and Stegun, I. A. ed.: *Handbook of Mathematical Functions*, Dover Publications, New York, 1965.

[2] 浅野：各種ソーティグアルゴリズムの実際的評価，情報処理学会アルゴリズム研究会 30-7, 1992.

[3] Cook, C.R. and Kim, D. J. : Best sorting algorithm for nearly sorted lists, *Comm. ACM*, Vol. 23, No. 11, (1980), pp. 620–624.

[4] Estivill-Castro and Wood : A survey of adaptive sorting algorithms, *ACM Comput. Surv*, Vol. 24, No. 4, (1992), pp. 441–476.

[5] 二村, 青木, 大谷, 白井: 整列法評価のためのランダム順列生成法, 情報処理学会アルゴリズム研究会, 44-3, 1995.

[6] 二村, 筧, 二村: 対称ヒープの実現とその応用, 情報処理学会アルゴリズム研究会, 28-7, 1992.

[7] 二村, 二村, 遠藤, 平井: 葉数最適整列法 LOAS とその実現法, 情報処理学会アルゴリズム研究会, 44-2, 1995.

[8] Graham, R.L., Knuth, D.E. and Patashnik, O. : *Concrete Mathematics*, Addison-Wesley, Reading, Massachusetts, 1989.

[9] Guputa, R., et al : On Randomization in Sequential and Distributed Algorithms, *ACM Computing Surveys*, Vol.26, No.1, (1994), pp. 7–86.

[10] Hoare, C.A.R: Quicksort, *Comput. J.*, ,Vol. 1, No. 1, (1962), pp. 10–15.

[11] Knuth, D.E. : *The Art of Computer Programming*, Vol. 1–3, Addison-Wesley, 1973.

[12] Manna, Z. : *Mathematical Theory of Computation*, McGRAW-HILL, New York, 1974.

[13] Sedgewick, R. : *Algorithms*, Addison-Wesley, 1989.

[14] Sprugnori, R. : The generation of binary trees as a numerical problem, *J. ACM*, Vol. 39, No. 2, (1992), pp. 317–327.

[15] Sprugnori, R. : Properties of binary trees related to position, *Comput. J.*, Vol. 35, No. 4, (1992), pp. 395–404.

[16] Stanley, R.P. : *Enumerative Combinatorics*, Vol. 1, Wadsworth & Books/Cole Advanced Books, Monterey, CA., 1986.

[17] Vuillemin, J : A unifying look at data structures, *Comm. ACM*, Vol. 23, No. 4, (1980), pp. 229–239.