

ランダムデータジェネレータジェネレータ (RDGG) の開発

3 J-2

二村良彦 青木健一 矢農正紀
早稲田大学 理工学部 情報学科

1 はじめに

アルゴリズムの精密な評価のためには、その性能に影響を及ぼす性質を制御しながらランダムに数列を生成し、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。しかし、指定された性質を持つ乱数列を生成することは容易ではない。

本研究の目的は、与えられた性質を有する乱順列を生成するランダムデータジェネレータを自動的に生成するランダムデータジェネレータジェネレータ (RDGG) を開発することである。本稿では先ず、乱数列の性質の指定方法を述べ、次にそれに基づき、ランダムデータジェネレータを自動生成する方法について報告する。

2 順列の単純指標

順列の生成法と用語の定義については文献 [1] で詳しく報告した。ここではジェネレータの説明に必要な用語について簡単に説明する。

順列の特性指標とは、順列 s から非負の整数上の関数または関数値である。そして、単純指標とは、要素を一つずつ挿入して順列を作成する過程で、指標の値が高々1ずつしか増加せず、しかも指標が増加する箇所と増加しない箇所の個数が、その時点の順列の長さ n と指標の値 m のみで決定されるような指標である。

単純指標の例としては、葉数 (左右より値が小さい要素の個数) や上昇部分 (左より右の値が大きい要素) の個数などがあげられる。

長さ n 単純指標 m の順列の総数 $S(n, m)$ の再帰部分は以下の式で表される。

$$S(n, m) = x(n, m)S(n-1, m) + y(n, m)S(n-1, m-1) \quad (1)$$

ここで $x(n, m)$ は、長さ $n-1$ 単純指標 m の順列に対し、指標を増加させずに長さ n にできる挿入箇所の個数を意味し、 $y(n, m)$ は、長さ $n-1$ 単純指標 $m-1$ の順列に対し、指標を1増加させて長さ n にできる挿入箇所の個数を意味する。

例えば、葉数の場合 $x(n, m) = 2m$, $y(n, m) = n - 2m + 2$ となり、上昇部分の場合 $x(n, m) = m + 1$, $y(n, m) = n - m$ となる。

指定された単純指標 m を持つ長さ n の順列をランダムに生成する場合、以下のようにすればよい。

1. 乱数を発生させて、確率 $\frac{x(n, m)S(n-1, m)}{S(n, m)}$ で2.へ、確率 $\frac{y(n, m)S(n-1, m-1)}{S(n, m)}$ で3.へ分岐する。
2. 指標 m 長さ $n-1$ の順列を再帰的に生成して、 $x(n, m)$ 箇所ある指標が増加しない場所のいずれかに、ランダムに要素を挿入する。
3. 指標 $m-1$ 長さ $n-1$ の順列を再帰的に生成し、 $y(n, m)$ 箇所ある指標が増加する場所のいずれかに、ランダムに要素を挿入する。

従って、構成比 $F(n, m) = \frac{S(n, m)}{x(n, m)S(n-1, m)}$ (確率の逆数) を計算することで、単純指標を持つ乱順列が生成できる。生成に要する時間は、構成比の計算に $O(mn)$ 時間、一つの順列の生成につき $O(n)$ 時間とできる。詳細については文献 [1] を参照されたい。

しかし、実際のプログラムでは核となる部分だけで数百ステップを要する。また、プログラミング上の工夫を凝らさないと計算量が $O(n)$ 倍増大する可能性を秘めている。従って、単純指標の仕様だけを与えて、その仕様を持つ乱順列のジェネレータが得られれば、一般プログラマにとっては有用であると考えられる。

ランダムデータジェネレータジェネレータ (以下 RDGG) は、個々の単純指標で違いのない部分のプログラミングを省略し、単純指標の性質を指定することで、容易にジェネレータを作成することを目的として開発された。

3 乱順列の仕様の記述方法

我々は、単純指標の仕様を記述したファイルを入力とし、C言語によるジェネレータを出力するようにRDGGを実装した。例えば、葉数の場合は次のように仕様を記述する。

```
%x 2*m
%y n - 2*m + 2
%0 m < 1 || n < 2*m - 1
%acp n..1
%inc (unode(L) && !leaf(R)) ||
      (unode(R) && !leaf(L))
%stay leaf(L) || leaf(R)
%fast
```

上昇部分の場合は次のように記述する。

```
%x m + 1
%y n - m
%0 m < 0 || n - 1 < m
%acp 1..n
%inc s(L) > s(R)
%stay s(L) < s(R)
%fast
```

仕様の定義は%で始まる疑似命令を用いて行う。式を記述する際の文法、演算子はC言語に従う。以下にその記述方法の概略を示す。

x,y 式(1)中の $x(n,m)$ と $y(n,m)$ を指定する。

0 指標の値が0になる条件を指定する。

acp 随伴順列(順列を生成するとき挿入していく要素の順列)を指定する。昇順の場合は1..nと、降順の場合はn..1とする。その他の場合は数式もしくは関数で指定できる。

inc,stay 要素を挿入することで指標が増える場合の条件をincで指定し、指標が増えない場合の条件をstayで指定する。条件の指定は次のような関数を用いて行う。

s 順列の要素の値を返す。

leaf 要素が葉(左右より値が小さい要素)であれば左から数えた番号を返し、そうでなければ0を返す。

unode 要素が一次節(左右どちらかより値が小さい要素)であれば左から数えた番号を返し、そうでなければ0を返す。

関数の引数には挿入箇所の付近の要素を指定できる。その対応を以下に示す。



その他の要素については数式で指定する。なお、以上を用いた式で条件を指定できない場合は、関数を記述することができる。

fast RDGGが出力するジェネレータは、標準では乱順列の生成に $O(n^2)$ の時間を必要とするが、要素の挿入後、挿入箇所の付近以外で指標の増加の有無が変化しない場合については、%fastを指定することで $O(n)$ へ改善できる。

さらに我々は、インターネットを通じて単純指標を持つ乱順列を入手できるようにした。その場合は、上記と同様の内容をWWW(World Wide Web)上で入力すればよい。

4 おわりに

RDGGの開発により、単純指標を持つ乱順列のジェネレータを統一的手順で容易に作成することが可能となった。なお、RDGGにはインターネット(<http://www.futamura.info.waseda.ac.jp/index-j.html>)を通じて誰でもアクセス可能である。それにより、指定された単純指標と長さを持つ乱順列を、指定された個数得ることができる。ただし、現在のRDGGは汎用性を持っているために1つの乱順列の生成には、例外を除いては $O(n^2)$ 時間およびスペースを要するので、長さが1000を超えるような順列の生成を行う場合には、講演番号3J-03のランダムデータサーバーを利用されたい。今後の課題は順列の指標として単純指標が属するクラスの解明と、長さ n が大きい際の近似計算へのRDGGの対応である。

参考文献

- [1] 二村, 大谷, 青木, 二村: 単純指標を持つ順列の高速生成法, 情報処理学会アルゴリズム研究会, 1997年1月。