Novedades de software/New softwares

ALGORITMO PARA LA GENERACIÓN ALEATORIA DE MATRICES BOOLEANAS INVERSIBLES

P. Freyre*, N. Díaz* , E. R. Morgado**
*Facultad de Matemática y Computación, Universidad de La Habana, Cuba.
**Facultad de Matemática, Física y Computación, Universidad Central "Marta Abreu" de las Villas. Cuba.

ABSTRACT
In the present paper we show a new algorithm for random generation of boolean square invertible matrices nxn. This algorithm has, as initial parameter, a randomly selected boolean matrix $A=\{a_{i,j}\}_{nxn}$, which has as the only restriction that there is not any $i\in \{1\ldots n\}$, such that $a_{i,i} = a_{i,i+1} = \ldots a_{i,n} = 0$. The program is presented in the Mathematic language.

MSC : 15A52

KEY WORDS: Vector – matrix products, Random matrix over finite fields.

RESUMEN
En el presente artículo mostramos un nuevo algoritmo para la generación aleatoria de matrices booleanas cuadradas nxn e inversibles. Este algoritmo tiene, como parámetro de entrada, una matriz booleana $A=\{a_{i,j}\}_{nxn}$, cuyos componentes se seleccionan aleatoriamente y que tiene, como única restricción, que no exista $i \in \{1\ldots n\}$ tal que $a_{i,i} = a_{i,i+1} = \ldots a_{i,n} = 0$. El algoritmo se expone programado en lenguaje Mathemática.

1. INTRODUCCIÓN

En Freyre P, Díaz N y Morgado E. R. (2009) se presenta en forma de pseudo código tres algoritmos para matrices cuadradas nxn con sus elementos pertenecientes a un campo finito GF(q), q potencia de un primo p, el primer Algoritmo permite la generación aleatoria de matrices, con el segundo se puede multiplicar un vector binario por una matriz seleccionada aleatoriamente y con el tercero se puede multiplicar un vector binario por la matriz inversa de una seleccionada aleatoriamente. En este trabajo también se expone el soporte teórico de estos algoritmos y se finaliza con un análisis de la complejidad de los mismos demostrándose que para el primer algoritmo la complejidad es menor que $O(n^3 \log n \log \log n)$ y para los dos restantes es menor que $O(n^2 \log n \log \log n)$.

En el presente trabajo se muestran programados, en lenguaje Mathemática, estos tres algoritmos para matrices booleanas inversibles nxn. Todos ellos utilizan al menos $n^2 - 1$ números aleatorios pertenecientes al campo binario $Z_2 =\{0,1\}$ y las operaciones a realizar son multiplicaciones de polinomios con coeficientes en Z_2, módulo un polinomio primitivo. Los dos últimos algoritmos tienen la ventaja, con relación a otros conocidos, de que, para efectuar la multiplicación de un vector por la matriz booleana inversible, seleccionada aleatoriamente, o por su inversa, no es necesario expresar explícitamente la matriz.

Un algoritmo estándar para generar de manera aleatoria una matriz no singular, con componentes en el campo Z_2, es aquel que selecciona aleatoriamente los n^2 elementos de la matriz y después calcula su determinante. El algoritmo se repite hasta que el determinante no sea congruente con cero mod 2. Si denotamos por E(n) el número esperado de veces que se repite el proceso, tenemos que $E(n) \leq 4$ (ver Pak I. (1997)). Este algoritmo necesita al menos n^2 bits aleatorios y tiene una complejidad menor que $O(n^3)$ (ver Knuth E. D. (1981)).

pfreyre@matcom.uh.cu

En Randal D. (1993) se presenta un algoritmo para generar de manera aleatoria una matriz booleana no singular. El tiempo esperado en la corrida del algoritmo es $M(n)+O(n^2)$, donde M(n) es el tiempo necesario para multiplicar dos matrices booleanas nxn, y el número esperado de bits aleatorios que utiliza el algoritmo es de n^2+3.

En el presente trabajo se muestran tres ejemplos de cálculo de una matriz aleatoria y de la multiplicación de un vector binario X por ésta y por su inversa.

2. GENERACIÓN ALEATORIA DE MATRICES BOOLEANAS

En este epígrafe se exponen programados en lenguaje Mathemática, los tres algoritmos anteriormente mencionados, en todos ellos se realiza como operación fundamental la multiplicación de dos polinomios f(x) y h(x) módulo g(x), f(x), h(x) y g(x) $\in Z_2[x]$. El polinomio g(x) es un polinomio primitivo arbitrario por lo que es más conveniente utilizar los que posean un número mínimo de coeficientes.

En Peterson W.W. y Weldon J. E. (1972), Golomb W. S. (1982) y Lidl R. y Niederreiter H. (1994) se encuentra información sobre los conceptos de polinomio irreducible y de polinomio primitivo y tablas con polinomios primitivos con coeficientes en el campo Z_2, de las cuales se pueden seleccionar los polinomios para estos algoritmos.

En los programas tenemos que:

n – Es el tamaño de la matriz.
lpp – Es la lista de polinomios primitivos a utilizarse en el algoritmo, representados en forma descendente según su grado, y se calculan con anterioridad. Los mismos pueden ser seleccionados arbitrariamente.
vbc – Es un parámetro de entrada y no es más que la matriz booleana $A=\{a_{i,j}\}_{nxn}$ expresada en forma de filas comenzando por la n – ésima hasta la primera. Los componentes de la matriz A se seleccionan aleatoriamente y tienen como única restricción, que no exista $i \in \{1\ldots n\}$ tal que $a_{i,i} = a_{i,i+1} = . . . a_{i,n} = 0$.
m – Es la matriz resultante.
v – Vector binario que se multiplica por la matriz.
vec – Vector resultante.

ALGORITMO PARA GENERAR DE FORMA ALEATORIA UNA MATRIZ BOOLEANA.

Programación del algoritmo.

```
Clear[Lbi]
Lbi[n_,i_,v_,vbc_,lpp_]:=
  Block[{x,t},
    x=lpp[[1]][Take[v,{1,i-1}]] +
        lpp[[1]][Take[vbc[[i]],{1,i-1}]]*lpp[[1]][Take[v,{i,i}]];
    If[TrueQ[x==0],x=lpp[[1]][PadLeft[{},n]]];
    t=lpp[[i]][Take[v,{i,n}]]*lpp[[i]][Take[vbc[[i]],{i,n}]];
    If[TrueQ[t==0],t=lpp[[i]][PadLeft[{},n+1-i]]];
    Return[Join[Take[x[[1]],{1,i-1}],t[[1]]]];
  ]
Clear[Genmatriz]
Genmatriz[n_,vbc_,lpp_]:=
  Block[{m={},v=IdentityMatrix[n],i,j,vec},
    For[j=1, j<=n,j++,
      i=j+1;vec=v[[j]];While[(i=i-1)>0,vec=Lbi[n,i,vec,vbc,lpp]];
      AppendTo[m,vec]
   ];
   Return[m];
 ]
```

ALGORITMO PARA LA MULTIPLICACIÓN DE UN VECTOR BINARIO X POR UNA MATRIZ SELECCIONADA ALEATORIAMENTE.

Programación del algoritmo.

```
Clear[Lbi]
Lbi[n_,i_,v_,vbc_,lpp_]:=
  Block[{x,t},
    x=lpp[[1]][Take[v,{1,i-1}]] +
        lpp[[1]][Take[vbc[[i]],{1,i-1}]]*lpp[[1]][Take[v,{i,i}]];
```

```
    If[TrueQ[x==0],x=lpp[[1]][PadLeft[{},n]]];
    t=lpp[[i]][Take[v,{i,n}]]*lpp[[i]][Take[vbc[[i]],{i,n}]];
    If[TrueQ[t==0],t=lpp[[i]][PadLeft[{},n+1-i]]];
    Return[Join[Take[x[[1]],{1,i-1}],t[[1]]]];
    ]
Clear[Mulvec]
Mulvec[n_,v_,vbc_,lpp_]:=
  Block[{vec},
    vec=v;Do[vec=Lbi[n,n - i,vec,vbc,lpp],{i,0,n-1}];
    Return[vec];
    ]
```

ALGORITMO PARA LA MULTIPLICACIÓN DE UN VECTOR BINARIO X POR LA INVERSA DE UNA MATRIZ SELECCIONADA ALEATORIAMENTE.

Programación del algoritmo.

```
Clear[ILb]
ILb[n_,i_,v_,vbc_,lpp_]:=
  Block[{x,t},
    t=lpp[[i]][Take[v,{i,n}]]*(lpp[[i]][Take[vbc[[i]],{i,n}]]^-1);
    If[TrueQ[t==0],t=lpp[[i]][PadLeft[{},n+1-i]]];
    x=lpp[[1]][Take[v,{1,i-1}]] -
       lpp[[1]][Take[vbc[[i]],{1,i-1}]]*lpp[[1]][t[[1]][[{1}]]];
    If[TrueQ[x==0],x=lpp[[1]][PadLeft[{},n]]];
    Return[Join[Take[x[[1]],{1,i-1}],t[[1]]]];
    ]
 Clear[IMulvec]
IMulvec[n_,v_,vbc_,lpp_]:=
  Block[{i,vec},
    i=0;vec=v;While[(i=i+1)<n+1,vec=ILb[n,i,vec,vbc,lpp]];
    Return[vec];
    ]
```

Ejemplo 1. Dados los polinomios primitivo: $1+x+x^6$; $1+x^2+x^5$; $1+x+x^4$; $1+x+x^3$; $1+x+x^2$; $1+x$, y los vbc= {{1,1,1,0,0,1},{1,1,0,0,1,1},{0,0,1,1,1,0}, {0,0,0,1,0,1},{0,1,0,1,1,0},{0,0,1,1,0,1}} que han sido seleccionados aleatoriamente.

Generación de la matriz aleatoria.

```
<<Algebra`FiniteFields`
lpp={GF[2,{1,1,0,0,0,0,1}],GF[2,{1,0,1,0,0,1}],
     GF[2,{1,1,0,0,1}],GF[2,{1,1,0,1}],
     GF[2,{1,1,1}],GF[2,{1,1}]}
vbc={{1,1,1,0,0,1},{1,1,0,0,1,1},{0,0,1,1,1,0},
    {0,0,0,1,0,1},{0,1,0,1,1,0},{0,0,1,1,0,1}}
m=Genmatriz[6,vbc,lpp]
MatrixForm[%]
```

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Multiplicación de un vector por la matriz.

```
<<Algebra`FiniteFields`
x={1,0,0,0,1,1}
y=Mulvec[6,x,vbc,lpp]
Y={0,0,1,0,1,1}
```

Multiplicación de un vector por la matriz inversa.

```
<<Algebra`FiniteFields`
x={1,0,0,0,1,1}
```

y=IMulvec[6,x,vbc,lpp]
Y={0,0,1,0,0,0}

Ejemplo 2. Dados los polinomios primitivo: $1+x^4+x^9$; $1+x^2+x^3+x^4+x^8$; $1+x^3+x^7$; $1+x+x^6$; $1+x^2+x^5$; $1+x+x^4$; $1+x+x^3$; $1+x+x^2$; $1+x$, y los vbc = {{0,1,1,1,1,1,0,0,1}, {1,0,1,1,1,1,1,1,0},{0,1,1,1,1,1,0,1,0}, {1,0,1,1,0,0,0,0,1},{0,1,0,1,1,0,1,0,0}, {1,0,1,1,1,0,0,0,1}, {0,1,0,1,0,1,0,0,1},{1,0,0,0,1,1,0,1,1},{0,0,1,1,1,0,0,0,1}} que han sido seleccionados aleatoriamente.

Generación de la matriz aleatoria.
<<Algebra`FiniteFields`
lpp = {GF[2,{1,0,0,0,1,0,0,0,0,1}],
GF[2,{1,0,1,1,1,0,0,0,1}],GF[2,{1,0,0,1,0,0,0,1}],
GF[2,{1,1,0,0,0,0,1}],GF[2,{1,0,1,0,0,1}],
GF[2,{1,1,0,0,1}],GF[2,{1,1,0,1}],GF[2,{1,1,1}],
GF[2,{1,1}]}
vbc = {{0,1,1,1,1,1,0,0,1},{1,0,1,1,1,1,1,1,0},
{0,1,1,1,1,1,0,1,0},{1,0,1,1,0,0,0,0,1},
{0,1,0,1,1,0,1,0,0},{1,0,1,1,1,0,0,0,1},
{0,1,0,1,0,1,0,0,1},{1,0,0,0,1,1,0,1,1},
{0,0,1,1,1,0,0,0,1}}
m = Genmatriz[9,vbc,lpp]
MatrixForm[%]

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Multiplicación de un vector por la matriz.
<<Algebra`FiniteFields`
x={0,1,1,0,0,1,1,0,1}
y=Mulvec[9,x,vbc,lpp]
Y={0,0,0,1,0,0,1,0,0}

Multiplicación de un vector por la matriz inversa.
<<Algebra`FiniteFields`
x={0,1,1,0,0,1,1,0,1}
y=IMulvec[9,x,vbc,lpp]
Y={0,0,0,0,0,0,0,0,1}

Ejemplo 3. Dados los polinomios primitivo: $1+x+x^6+x^{10}+x^{14}$; $1+x+x^3+x^4+x^{13}$; $1+x+x^4+x^6+x^{12}$; $1+x^2+x^{11}$; $1+x^3+x^{10}$; $1+x^4+x^9$; $1+x^2+x^3+x^4+x^8$; $1+x^3+x^7$; $1+x+x^6$; $1+x^2+x^5$; $1+x+x^4$; $1+x+x^3$; $1+x+x^2$; $1+x$, y los vbc = {{1,1,0,1,0,1,0,1,1,1,0,0,0,1}, {0,0,1,0,1,0,1,0,1,0,1,1,0,1}, {1,1,1,0,0,1,0,0,1,0,0,1,0,0}, {0,1,1,1,1,0,1,1,1,1,1,1,1,0}, {1,1,0,0,1,1,0,1,1,0,0,0,0,1}, {1,1,1,1,0,1,1,1,1,0,0,0,1,0}, {1,0,0,1,0,1,1,0,0,1,0,0,0,1}, {1,1,0,1,0,0,0,0,0,0,1,1,0,1}, {0,0,0,0,0,0,1,0,1,1,1,0,0,0}, {1,1,1,1,1,0,0,1,0,0,0,1,0,1}, {1,0,0,1,0,1,0,1,0,0,1,1,1,1}, {1,1,0,0,0,1,1,1,0,0,0,0,1,0}, {0,0,0,0,0,1,0,0,0,1,0,0,1,0} ,{0,0,1,1,0,1,0,0,1,1,1,1,0,1}} que han sido seleccionados aleatoriamente.

Generación de la matriz aleatoria.
<< "Algebra`FiniteFields`"

```
lpp = {GF[2,{1,1,0,0,0,0,1,0,0,0,1,0,0,0,1}],
      GF[2,{1,1,0,1,1,0,0,0,0,0,0,0,0,1}],
      GF[2,{1,1,0,0,1,0,1,0,0,0,0,0,1}],
      GF[2,{1,0,1,0,0,0,0,0,0,0,0,1}],
      GF[2,{1,0,0,1,0,0,0,0,0,0,1}],
      GF[2,{1,0,0,0,1,0,0,0,0,1}],
      GF[2,{1,0,1,1,1,0,0,0,1}],GF[2,{1,0,0,1,0,0,0,1}],
      GF[2,{1,1,0,0,0,0,1}],GF[2,{1,0,1,0,0,1}],
      GF[2,{1,1,0,0,1}],GF[2,{1,1,0,1}],GF[2,{1,1,1}],
      GF[2,{1,1}]}
vbc = {{1,1,0,1,0,1,0,1,1,1,0,0,0,1},
      {0,0,1,0,1,0,1,0,1,0,1,1,0,1},
      {1,1,1,0,0,1,0,0,1,0,0,1,0,0},
      {0,1,1,1,1,0,1,1,1,1,1,1,1,0},
      {1,1,0,0,1,1,0,1,1,0,0,0,0,1},
      {1,1,1,1,0,1,1,1,1,0,0,0,1,0},
      {1,0,0,1,0,1,1,0,0,1,0,0,0,1},
      {1,1,0,1,0,0,0,0,0,0,1,1,0,1},
      {0,0,0,0,0,0,1,0,1,1,1,0,0,0},
      {1,1,1,1,1,0,0,1,0,0,0,1,0,1},
      {1,0,0,1,0,1,0,1,0,0,1,1,1,1},
      {1,1,0,0,0,1,1,1,0,0,0,0,1,0},
      {0,0,0,0,0,1,0,0,0,1,0,0,1,0},
      {0,0,1,1,0,1,0,0,1,1,1,1,0,1}}
m = Genmatriz[14, vbc, lpp]
MatrixForm[%]
```

$$
\begin{pmatrix}
1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Multiplicación de un vector por la matriz.

```
<<Algebra`FiniteFields`
x={0,0,1,1,0,1,0,0,1,1,1,1,0,1}
y=Mulvec[14,x,vbc,lpp]
Y={1,0,1,0,1,1,0,1,1,1,1,0,0,1}
```

Multiplicación de un vector por la matriz inversa.

```
<<Algebra`FiniteFields`
x={0,0,1,1,0,1,0,0,1,1,1,1,0,1}
y=IMulvec[14,x,vbc,lpp]
```

Y={0,1,1,0,0,1,1,0,1,1,1,0,0,0}

RECEIVED APRIL, 2008
REVISED MARCH 2010

REFERENCIAS

[1] FREYRE P., DÍAZ N. Y MORGADO E. R. **(2009):** Fast algorithm for the multiplication of a row vector by a randomly selected matrix A. **Journal of Discrete Mathematical Sciences & Cryptography,** 12, 533–549.

[2] GOLOMB W. S. (1982). **Shift Register Sequences**. Aegean Park Press. California.

[3] KNUTH E. D. (1981). **The Art of Computer Programming**. Vol 2. Addison – Wesley. 2da ed. , N. York.

[4] PAK I. (1997). **Random Walks on Groups: Strong Uniform Time Approach**. (http://www-math.mit.edu).

[5] RANDAL D. (1993). **Efficient Generation of Random Nonsingular Matrices**. (http://citeseer.ist.psu.edu).

[6] LIDL R. y NIEDERREITER H. (1994). **Introduction to Finite Fields and their Applications**. Cambridge University. New York.

[7] PETERSON W.W. y WELDON J. E. (1972). **Error – Correcting Codes,** 2ed.. John Wiley and Sons, Inc. New York.