

PROCESSO FLIP-MATANÇA-MITOSIS

Samuel Victor Medeiros de Macêdo¹; André Toom²

¹Estudante do Curso de estatística -DE -UFPE; E-mail: dmmada@hotmail.com, ²Docente/pesquisador do Depto de estatística - DE - UFPE. E-mail: toom@feller.de.ufpe.br

RESUMO

Flip-Matança-Mitosis consiste num processo unidimensional no tempo discreto com iteração local, onde componentes podem desaparecer e reaparecer ao longo do processo de evolução. Nosso projeto tem duas finalidades, a primeira é escrever pseudo-códigos para o processo Flip-Matança-Mitosis, pois dessa forma fica mais simples o entendimento do algoritmo, a segunda é que iremos propor um novo algoritmo que simula mais rápido que o tradicional.

Iremos aqui focar os processos do tipo circular, pois são mais viáveis para programação.

DEFINIÇÕES

Definimos aqui x como a posição em que o componente se encontra onde x pertence a Z e $e(x)$ como o estado onde $e(x)$ pertence ao conjunto $\{MENOS,MAIS\}$.

Ocorrem a cada instante de tempo três funções: Flip, Matança e Mitosis. A primeira função é a Flip que consiste que um componente muda o estado de menos pra mais ou de mais para menos, com probabilidade β , a segunda é a função Matança que, se dois componentes tiverem sinais diferentes eles se matam com probabilidade α , e por ultimo é a função Mitosis que duplica um componente com probabilidade γ . Se um componente for "morto" pela função matança ele recebe um pseudo estado chamado morte e não será mais lido pelo algoritmo.

FUNÇÃO FLIP(β)

```
Gere U(0,1)
Se (U <  $\beta$ )
{
  se (e(x) = MAIS)
    e(x) ← MENOS
  senão
    e(x) = MENOS
}
```

FUNÇÃO MATANÇA(α)

```
Gere U(0,1)
Se (U <  $\alpha$ )
{
  se (e(x) = MENOS)
    se (e(x+1) = MAIS)
    {
      e(x+1) ← morte
      e(x) ← morte
    }
  senão
    se (e(x+1) = menos)
    {
      e(x+1) ← morte
      e(x) ← morte
    }
}
```

FUNÇÃO MITOSIS(γ)

```
Gere U(0,1)
Se (U <  $\gamma$ )
{
  se (e(x) = MENOS)
    e(x) ← MENOS,MENOS
  senão
    e(x) ← MAIS, MAIS
}
```

PROCESSO FLIP-MATANÇA-MITOSIS

O processo inicia com todos menos, passa-se a função Flip por todos os estados, um a um, mudando-se o sinal ou não, depois passa-se a função matança por todos e por ultimo a função mitosis. A outra forma que queremos propor para esse processo é aleatorizando as chamadas da seguinte forma: suponha α, β, γ como taxas e ao invés de passar as funções uma de cada vez por todos os componentes do processo gerando sempre uma $U(0,1)$, geramos uma $U(0,V)$ onde $V = \alpha + \beta + \gamma$ e daí já verificamos qual função vamos usar nesse componente. Definindo que o processo seja circular, x como a posição em que o componente se encontra e $e(x)$ como o estado, então podemos escrever esses pseudo-códigos dessa forma:

FORMA CLÁSSICA

```
Faça
{
  Faça
  {
    Função Flip( $\beta$ )
    x ← x+1
  }
  Até completar uma volta
  Faça
  {
    Função Matança( $\alpha$ )
    x ← x+1
  }
  Até completar uma volta
  Faça
  {
    Função Mitosis( $\gamma$ )
    x ← x+1
  }
  Até completar uma volta
Até todos MENOS
}
```

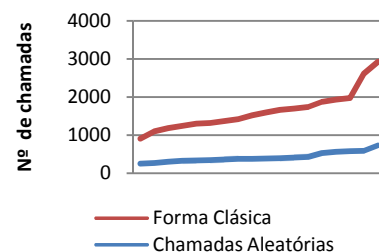
CHAMADAS ALEATÓRIAS

```
V =  $\alpha + \beta + \gamma$ 
Faça
{
  Gere U(0,V)
  Caso (V <  $\alpha$ )
    Chame função Matança( $\alpha$ )
  Caso ( $\alpha < V < \alpha + \beta$ )
    Chame função Flip( $\beta$ )
  Caso ( $\alpha + \beta < V < \alpha + \beta + \gamma$ )
    Chame função Mitosis( $\gamma$ )
    x ← x+1
  }
Até todos MENOS
```

RESULTADOS

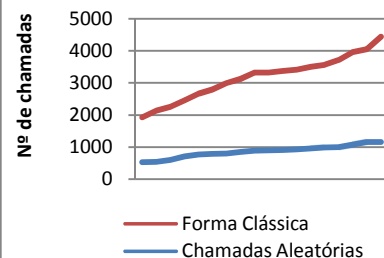
Foram feitas simulações com diversas quantidades de componentes iniciais, cada algoritmo foi gerado 1000 vezes para cada caso. O processo deve parar quando todos os componentes voltam para o estado $e(x) = MENOS$, então é de interesse saber se o algoritmo com chamadas aleatórias chegam ao mesmo resultado (todos MENOS), fazendo menos operações. Veremos a seguir alguns gráficos comparativos mostrando como os algoritmos se comportam em relação à quantidade de vezes em que uma alguma das funções é utilizada, ou seja, o número de chamadas de funções.

Análise para 50 componentes iniciais



É interessante notar também que com o aumento de componentes iniciais a diferença do número de chamadas funções entre os dois algoritmos é ainda maior.

Análise para 100 componentes iniciais



CONCLUSÕES

O algoritmo com chamadas aleatórias simula o processo Flip-Matança-Mitosis de uma forma melhor que o processo clássico. Além de evitar geração de variáveis aleatórias desnecessárias também apresenta uma diminuição de operações, pois chega ao fim do processo usando menos chamadas de funções.

BIBLIOGRAFIA

- A.Toom. *Particles Systems with Variables Length*. Bulletin of Brazillian Mathematical Society, 33(3), pp. 419-425, 2002
- A. Toom. *Non-Ergodicity in a 1-D Particle Process with variable Length*. Journal of Statistical Physics, 115(3/4), pp. 895-924, 2004
- A.D. Ramos. *Processos de Partículas com comprimento Variável*. Tese de Doutorado. Universidade Federal de Pernambuco, Departamento de estatística, Recife Pernambuco, Brasil, 2007
- B.R. James. *Probabilidade: um curso em nível intermediário*. Instituto de Matemática Pura e Aplicada, Rio de Janeiro, 2002