On the Cost of Generating PH-distributed Random Numbers

Philipp Reinecke, Katinka Wolter
{philipp.reinecke, katinka.wolter}@fu-berlin.de

Miklos Telek, Levente Bodrog
{telek, bodrog}@webspn.hit.bme.hu
PH-Distributions: Applications

- PH-distribution: Distribution of time to absorption in an MC
- Obtain PH models for e.g. response times using PH-FIT, G-FIT, moment-matching methods, etc.
- Use the PH models for analytic solutions:
  - e.g. M|PH|1
  - state-space explosion necessitates small models
- Use the PH models in simulations:
  - M|PH|1 with restart; fault-injection in testbeds
  - requires models that allow fast random-number generation
Elementary Operations and Cost Metrics

- **Elementary Operations:**
  
  Uniform random variate:  $U$
  
  Exponential random variate:  $\text{Exp}(\lambda) = -\frac{1}{\lambda} \ln U$
  
  Erlang random variate:  $\text{Erl}(b, \lambda) = -\frac{1}{\lambda} \ln \prod_{i=1}^{b} U_i$
  
  Geometric random variate on 0, 1, ...:  $\text{Geo}(p) = \left\lfloor \frac{\ln U}{\ln p} \right\rfloor$

- **Metrics:**
  
  - #uni: Number of uniform random variates
  
  - #ln: Number of logarithms
PH Classes

Hyper-Erlang Distributions

Acyclic PH Distributions

General PH Distributions

\( m \): Number of branches
\( \beta \): Branch probabilities
\( b \): Branch lengths
\( \lambda \): Rates
\( n = b \): Number of phases

\( \alpha \): Initial probabilities
\( \lambda \): Rates
\( n \): Number of phases

\( \alpha \): Initial probabilities
\( A \): Subgenerator matrix
\( n \): Number of phases
Algorithms and Costs

Hyper-Erlang Distributions ⊆ Acyclic PH Distributions ⊆ General PH Distributions

Obvious approach: Play the CTMC

\[ n^* = \alpha (-\text{Diag}(\frac{1}{a_{ii}})A)^{-1} \mathbf{1} \text{ states} \]

Costs:
\[ \#\text{uni} = 2n^* + 1 \]
\[ \#\text{ln} = n^* \]

Neuts/Pagano (1981) approach: Build Erlangs

\[ \sum_{j=1}^{k_i} \text{Exp}(\lambda_j) \text{ for } k_i \text{ visits to state } i \]

Replace by \( \text{Erl}(k_i, \lambda_i) \)

Costs:
\[ \#\text{uni} = 2n^* + 1 \]
\[ \#\text{ln} = n \]

\( \alpha \): Initial probabilities
\( A \): Subgenerator matrix
\( n \): Number of phases
Class-specific Algorithms and Costs

Hyper-Erlang
Distributions

ACyclic PH
Distributions

General PH
Distributions

$\begin{align*}
\mathbb{P} & : \text{Initial probabilities} \\
n & : \text{Number of phases}
\end{align*}$

$\begin{align*}
m & : \text{Number of branches} \\
\beta & : \text{Branch probabilities} \\
b & : \text{Branch lengths} \\
\lambda & : \text{Rates} \\
n = b_1 & : \text{Number of phases}
\end{align*}$

$\mathbb{P}$: Initial probabilities
$n$: Number of phases

**HErD Algorithm:**
Choose and draw Erlang sample
Traversed states: $n^* = \beta b^T$

Costs:
$\#uni = n^* + 1$
$\#ln = 1$

**CF-1 Algorithm:**
Do not draw random number for choice of next state.
Traversed states: $n^* = \alpha \nu^T$,
where $\nu = (n, n-1, ..., 1)$

Costs
$\#uni = n^* + 1$
$\#ln = n^*$
Similarity Transformations and Costs

Hyper-Erlang Distributions \( \subseteq \) Acyclic PH Distributions \( \subseteq \) General PH Distributions

Similarity Transformation:
Matrix \( B \) with \( B \mathbf{1}=\mathbf{1} \)
\( D=B^{-1}AB \)
\( \delta =\alpha B \)
\( (\alpha, A) \) and \( (\delta, D) \) are the same distribution

Example

\( \alpha =\left(0.7, 0.1, 0.2\right) \), \( A=\begin{pmatrix} -1 & 0.9 & 0.1 \\ 1.5 & -2 & 0.5 \\ 2.5 & 0 & -3 \end{pmatrix} \)

Similarity transformation with matrix \( B=\begin{pmatrix} 1.04 & 0 & -0.04 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \)

yields \( \delta =\left(0.728, 0.1, 0.172\right) \) and \( D=\begin{pmatrix} -0.9 & 0.865385 & 0.0153846 \\ 1.56 & -2 & 0.44 \\ 2.6 & 0 & -3.1 \end{pmatrix} \)

Costs

\( n^*(\alpha, A)=39.86 \)
\( n^*(\delta, D)=36.72 \)
Monocyclic Representations and Costs

General PH Distributions

= Monocyclic Representations

\[ \alpha : \text{Initial probabilities} \]
\[ A : \text{Subgenerator matrix} \]
\[ n : \text{Number of phases} \]

Example

\[ \alpha = (0.7, 0.1, 0.2), \quad A = \begin{pmatrix} -1 & 0.9 & 0.1 \\ 1.5 & -2 & 0.5 \\ 2.5 & 0 & -3 \end{pmatrix} \]

Monocyclic representation:

\[ \gamma = (0.945, 0.014, 0.004, 0.038), \quad G = \begin{pmatrix} -0.035 & 0.035 & 0 & 0 \\ 0 & -2.67 & 2.67 & 0 \\ 0 & 0 & -2.67 & 2.67 \\ 0 & 0.035 & 0 & -2.67 \end{pmatrix} \]

Costs

\[ n^*(\alpha, A) = 39.86 \]
\[ n^*(\gamma, G) = 3.90 \]
Monocyclic Algorithm and Costs

General PH Distributions

Monocyclic Algorithm

Choose initial state $i$ ($\alpha$-distributed)
State $i$ is in FE block $j$.
$l$ is the number of states between $i$ and the end of the block.
Draw the number of loops: $c = \text{Geo}(z_j)$
Draw random number for block $j$: $x_j = \text{Erl}(cb_j + l, \lambda_j)$
For each subsequent block $(k = j + 1, ..., m)$:
$$c_k = \text{Geo}(z_k)$$
$$x_k = \text{Erl}(c_kb_k, \lambda_k)$$
Return $\sum_{k=j}^{n} x_k$

Costs

$\#\text{uni} = n^* + 1 + \omega \varphi$
where $\omega_i$ is the probability of entering block $i$
and $\varphi = (m, m-1, ..., 1)$
$\#\text{ln} = 3 \omega \varphi^T$

$\alpha$: Initial state probabilities
$m$: Number of Feedback Erlang blocks
$b$: Length of FE blocks
$\lambda$: Rates of FE blocks
$z$: Feedback probabilities of FE blocks
$n = b1$: Number of phases
## Comparison

### Average Traversed States

\[ n^*(\alpha, A) = 39.86 \]

After similarity transformation:
\[ n^*(\delta, D) = 36.73 \]

With monocyclic representation:
\[ n^*(\gamma, G) = 3.90 \]

### Average Costs (t for \(10^7\) samples)

<table>
<thead>
<tr>
<th>Play ((\alpha, A))</th>
<th>Neuts/Pagano ((\alpha, A))</th>
<th>Neuts/Pagano ((\delta, D))</th>
<th>Monocyclic ((\gamma, G))</th>
</tr>
</thead>
</table>
| \#uni = 2n^* + 1 = 80.72 | \#uni = 2n^* + 1 = 80.72 | \#uni = 74.46 | \#uni = \(n^* + 1 + \omega \varphi^T = n^* + 1 + (0.95, 0.05) \begin{pmatrix} 2 \\ 1 \end{pmatrix} \)
| \#ln = n^* = 39.86 | \#ln = n = 3 | \#ln = 3 | = 6.85 |
| \(t = 217s\) | \(t = 155s\) | \(t = 142s\) | \#ln = 3 \(\omega \varphi^T\)
| \#uni = 74.46 | \#ln = 3 | \#ln = 4 | = 5.85 |
| \#ln = 36.73 | \(t = 196s\) | \(t = 23s\) | \(t = 22s\) |
Conclusion

- Costs for PH-PRNG depend on
  - choice of class
  - choice of algorithm
  - choice of representation

- For general PH, choice of representation appears promising

- Optimisation problem: Find a (not necessarily minimal) PH representation that minimises the average number of traversed states $n^*$. 
Fin.
Obvious Approach: Play the Markov Chain

- Choose initial state: $i=3$
- Draw $x_1=\text{Exp}(-3)$, choose next state: $i=1$
- Draw $x_2=\text{Exp}(-1)$, choose next state: $i=2$
- Draw $x_3=\text{Exp}(-2)$, choose next state: $i=3$
- Draw $x_4=\text{Exp}(-3)$, choose next state: $i=1$
- Draw $x_5=\text{Exp}(-1)$, choose next state: $i=3$
- Draw $x_6=\text{Exp}(-3)$, choose next state: $i=4$
- Return $x_1 + \ldots + x_6$
Costs for Playing the CTMC

• Worst case is undefined

• Average case:
  
  - traverse $n^* = \alpha (\text{Diag}(1/a_{ii}) A)^{-1} \mathbf{1}$ phases
  
  - draw one uniform for the initial state
  
  - draw one uniform for each state selection
  
  - draw $n^*$ exponential random variates
  
  - Average costs:

    • $#\ln = n^*$
    
    • $#\text{uni} = 2n^* + 1$
Neuts/Pagano (1981): Count State Traversals

- Return value of Play:
  \[ \text{Exp}(-3) + \text{Exp}(-1) + \text{Exp}(-2) + \text{Exp}(-3) + \text{Exp}(-1) + \text{Exp}(-3), \]
  which is equivalent to
  \[ \text{Exp}(-1) + \text{Exp}(-1) + \text{Exp}(-2) + \text{Exp}(-3) + \text{Exp}(-3) = \text{Erl}(2, -1) + \text{Erl}(1, -2), \text{Erl}(1, -3) \]

- Basic Idea: Count traversals, construct Erlangs

- Worst-case costs: \#ln=n (#uni = \infty)

- Average case costs:
  - \#ln = n
  - \#uni = 2n^* + 1
Use the monocyclic representation

- Monocyclic representation (Mocanu/Commault 1999):

\[0.95\ 0.014\ 0.0038\ 0.038\]

\[0.035\ 0.267\ z^*0.267\]

\[z=0.013\]
Monocyclic representation \((\mathbf{y}, \mathbf{G})\):
\[
\alpha = (0.95, 0.01, 0.004, 0.04)
\]
\[
m = 2
\]
\[
\mathbf{b} = (1, 4)
\]
\[
\lambda = (0.035, 2.67)
\]
\[
\mathbf{z} = (0, 0.013)
\]
\[
n = 4
\]