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: $\mathrm{Exp}(\lambda) = -\frac{1}{\lambda} \ln U$

  Erlang random variate: $\mathrm{Erl}(b, \lambda) = -\frac{1}{\lambda} \ln \prod_{i=1}^{b} U_i$

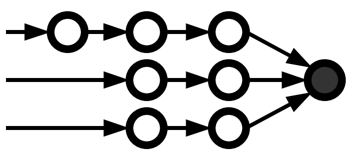
  Geometric random variate on 0, 1, ...: $\mathrm{Geo}(p) = \left\lfloor \frac{\ln U}{\ln p} \right\rfloor$

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

PH Classes

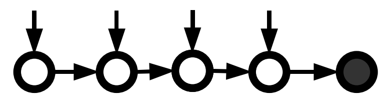
Hyper-Erlang Distributions $\subset$ Acyclic PH Distributions $\subset$ General PH Distributions

Hyper-Erlang Distributions



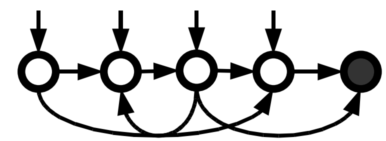
m: Number of branches
$\boldsymbol{\beta}$: Branch probabilities
$\boldsymbol{b}$: Branch lengths
$\boldsymbol{\lambda}$: Rates
$n = \boldsymbol{b}1$: Number of phases

Acyclic PH Distributions



$\boldsymbol{\alpha}$: Initial probabilities
$\boldsymbol{\lambda}$: Rates
n: Number of phases

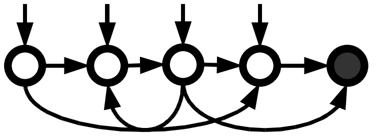
General PH Distributions



$\boldsymbol{\alpha}$: Initial probabilities
$\mathbf{A}$: Subgenerator matrix
n: Number of phases

Algorithms and Costs

Hyper-Erlang Distributions $\subset$ Acyclic PH Distributions $\subset$ General PH Distributions



Obvious approach: Play the CTMC

Traverses $n^* = \boldsymbol{\alpha} \left(-\mathrm{Diag}\langle \frac{1}{-a_{ii}} \rangle \mathbf{A}\right)^{-1} \mathbf{1}$ states

Costs:
$\#\mathrm{uni} = 2n^* + 1$
$\#\ln = n^*$

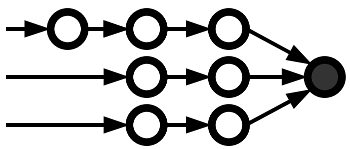
$\boldsymbol{\alpha}$: Initial probabilities
$\mathbf{A}$: Subgenerator matrix
n : Number of phases

Neuts/Pagano (1981) approach: Build Erlangs

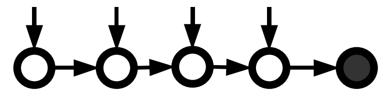
Replace $\sum_{j=1}^{j=k_i} \mathrm{Exp}(\lambda_i)$ for k_i visits to state i
by $\mathrm{Erl}(k_i, \lambda_i)$
Costs:
$\#\mathrm{uni} = 2n^* + 1$
$\#\ln = n$

Class-specific Algorithms and Costs

Hyper-Erlang Distributions $\subset$ Acyclic PH Distributions $\subset$ General PH Distributions



m : Number of branches
$\boldsymbol{\beta}$: Branch probabilities
$\boldsymbol{b}$: Branch lengths
λ : Rates
$n = \boldsymbol{b}1$: Number of phases



$\boldsymbol{\alpha}$: Initial probabilities
λ : Rates
n : Number of phases

HErD Algorithm:

Choose and draw Erlang sample
Traversed states: $n^* = \boldsymbol{\beta}\, \boldsymbol{b}^{\boldsymbol{T}}$

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

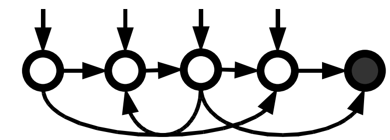
CF-1 Algorithm:

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

Costs
$\#\mathrm{uni} = n^* + 1$
$\#\ln = n^*$

Similarity Transformations and Costs

Hyper-Erlang Distributions $\subset$ Acyclic PH Distributions $\subset$ General PH Distributions



$\boldsymbol{\alpha}$: Initial probabilities
$\mathbf{A}$: Subgenerator matrix
n : Number of phases

Similarity Transformation:

Matrix $\mathbf{B}$ with $\mathbf{B}\mathbf{1}=\mathbf{1}$
$\mathbf{D}=\mathbf{B}^{-1}\mathbf{A}\mathbf{B}$
$\boldsymbol{\delta}=\boldsymbol{\alpha}\mathbf{B}$
$(\boldsymbol{\alpha},\mathbf{A})$ and $(\boldsymbol{\delta},\boldsymbol{D})$ are the same distribution

Example

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

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

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

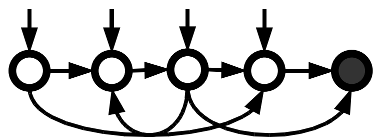
Costs

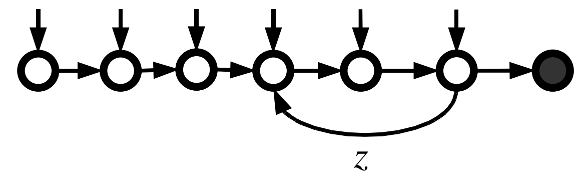
$n^*(\boldsymbol{\alpha},\mathbf{A})=39.86$

$n^*(\boldsymbol{\delta},\mathbf{D})=36.72$

Monocyclic Representations and Costs

General PH Distributions = Monocyclic Representations





$\boldsymbol{\alpha}$: Initial probabilities
$\mathbf{A}$: Subgenerator matrix
n : Number of phases

$\boldsymbol{\alpha}$: Initial state probabilities
m : Number of Feedback Erlang blocks
$\boldsymbol{b}$: Length of FE blocks
$\boldsymbol{\lambda}$: Rates of FE blocks
$\boldsymbol{z}$: Feedback probabilities of FE blocks
$n = \boldsymbol{b}\mathbf{1}$: Number of phases

Example

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

Monocyclic representation:

$$\boldsymbol{\gamma} = (0.945, 0.014, 0.004, 0.038),$$

$$\mathbf{G} = \begin{pmatrix} -0.035 & 0.035 & 0 & 0 \\ 0 & -2.67 & 2.67 & 0 \\ 0 & 0 & -2.67 & 2.67 \\ 0 & 0.035 & 0 & -2.67 \end{pmatrix}$$

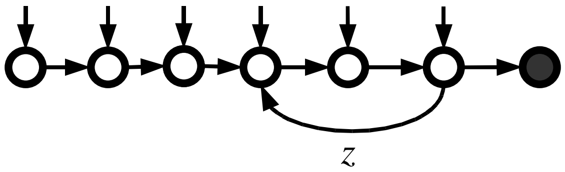
Costs

$$n^*(\boldsymbol{\alpha}, \mathbf{A}) = 39.86$$

$$n^*(\boldsymbol{\gamma}, \mathbf{G}) = 3.90$$

Monocyclic Algorithm and Costs

General PH Distributions = Monocyclic Representations



Monocyclic Algorithm

Choose initial state i (α − 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 $(k = j+1, \ldots, m)$:
$c_k = \text{Geo}(z_k)$
$x_k = \text{Erl}(c_k b_k, \lambda_k)$
Return $\sum_{k=j}^{n} x_k$

$\boldsymbol{\alpha}$: Initial state probabilities
m: Number of Feedback Erlang blocks
$\boldsymbol{b}$: Length of FE blocks
$\boldsymbol{\lambda}$: Rates of FE blocks
$\boldsymbol{z}$: Feedback probabilities of FE blocks
$n = \boldsymbol{b}\mathbf{1}$: Number of phases

Costs

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

Comparison

Average Traversed States

$n^*(\boldsymbol{\alpha}, \mathbf{A}) = 39.86$

After similarity transformation:

$n^*(\boldsymbol{\delta}, \mathbf{D}) = 36.73$

With monocyclic representation:

$n^*(\boldsymbol{\gamma}, \mathbf{G}) = 3.90$

Average Costs (t for 10^7 samples)

Play $(\boldsymbol{\alpha}, \mathbf{A})$:

$\#\text{uni} = 2\text{n}^* + 1 = 80.72$

$\#\ln = n^* = 39.86$

$t = 217\text{s}$

Neuts/Pagano$(\boldsymbol{\alpha}, \mathbf{A})$:

$\#\text{uni} = 2\text{n}^* + 1 = 80.72$

$\#\ln = n = 3$

$t = 155\text{s}$

Play$(\boldsymbol{\delta}, \mathbf{D})$:

$\#\text{uni} = 74.46$

$\#\ln = 36.73$

$t = 196\text{s}$

Neuts/Pagano$(\boldsymbol{\delta}, \mathbf{D})$:

$\#\text{uni} = 74.46$

$\#\ln = 3$

$t = 142\text{s}$

Play$(\boldsymbol{\gamma}, \mathbf{G})$:

$\#\text{uni} = 8.8$

$\#\ln = 3.90$

$t = 21\text{s}$

Neuts/Pagano$(\boldsymbol{\gamma}, \mathbf{G})$:

$\#\text{uni} = 8.8$

$\#\ln = 4$

$t = 23\text{s}$

Monocyclic$(\boldsymbol{\gamma}, \mathbf{G})$:

$\#\text{uni} = n^* + 1 + \boldsymbol{\omega}\, \boldsymbol{\varphi}^T = n^* + 1 + (0.95, 0.05)\begin{pmatrix}2\\1\end{pmatrix}$

$= 6.85$

$\#\ln = 3\boldsymbol{\omega}\, \boldsymbol{\varphi}^T$

$= 5.85$

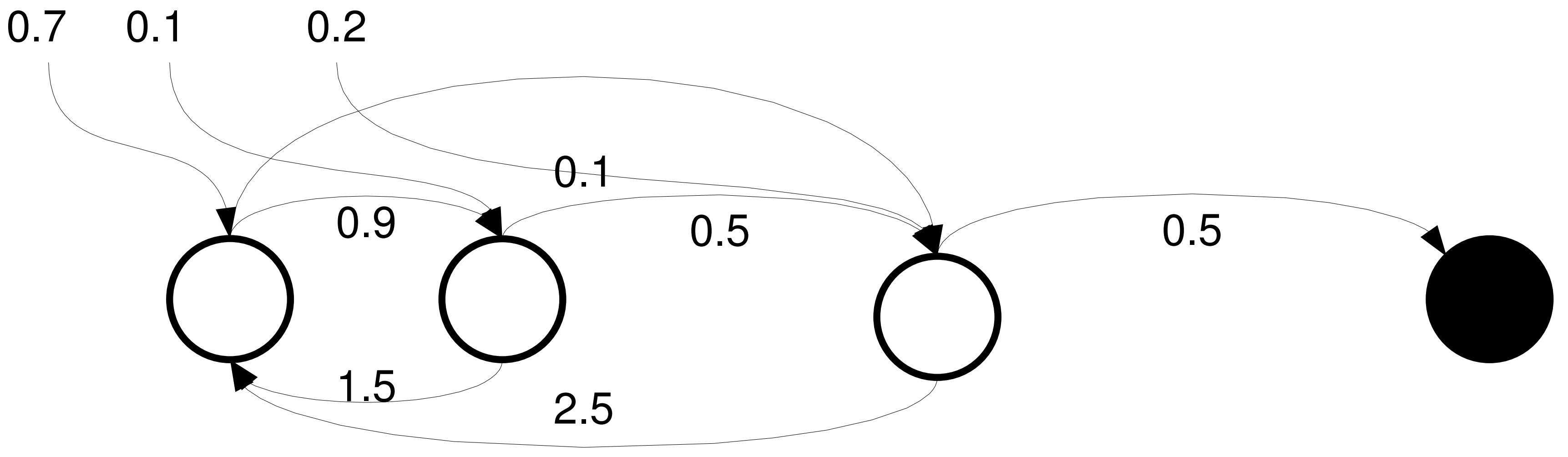
$t = 22\text{s}$

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 x1=Exp(-3), choose next state: i=1
- Draw x2=Exp(-1), choose next state: i=2
- Draw x3=Exp(-2), choose next state: i=3
- Draw x4=Exp(-3), choose next state: i=1
- Draw x5=Exp(-1), choose next state: i=3
- Draw x6=Exp(-3), choose next state: i=4
- Return x1 + ... + x6

Costs for Playing the CTMC

- Worst case is undefined
- Average case:
  - traverse $n^* = \alpha\,(Diag\langle 1/a_{ii}\rangle \boldsymbol{A})^{-1}\mathbf{1}$ phases
  - draw one uniform for the initial state
  - draw one uniform for each state selection
  - draw n* exponential random variates
  - Average costs:
    - #ln = n*
    - #uni = 2n* + 1

Neuts/Pagano (1981): Count State Traversals

- Return value of Play:

  Exp(-3) + Exp(-1) + Exp(-2) + Exp(-3) + Exp(-1) + Exp(-3),

  which is equivalent to

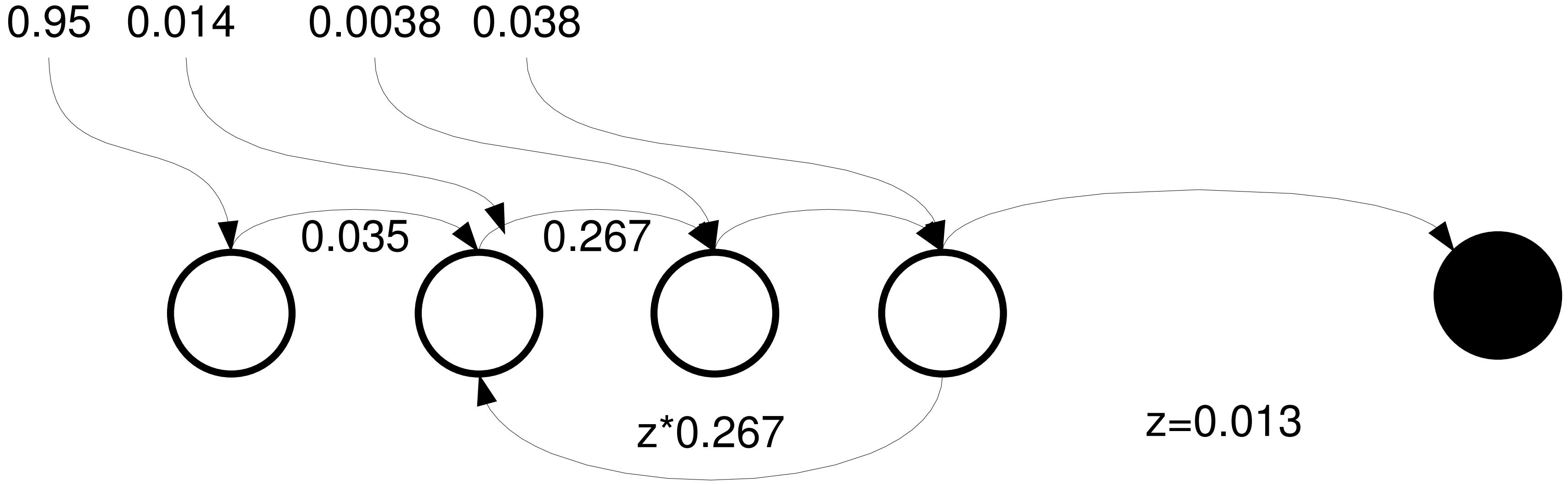
  Exp(-1) + Exp(-1) + Exp(-2) + Exp(-3) + Exp(-3)

  = Erl(2, -1) + Erl(1, -2), Erl(1, -3)

- Basic Idea: Count traversals, construct Erlangs
- Worst-case costs: #ln=n (#uni = ∞)
- Average case costs:
  - #ln = n
  - #uni = 2n* + 1

Use the monocyclic representation

- Monocyclic representation (Mocanu/Commault 1999):



Monocyclic representation $(\boldsymbol{\gamma}, \boldsymbol{G})$:

$\boldsymbol{\alpha} = (0.95, 0.01, 0.004, 0.04)$

$m = 2$

$\boldsymbol{b} = (1, 4)$

$\boldsymbol{\lambda} = (0.035, 2.67)$

$\boldsymbol{z} = (0, 0.013)$

$n = 4$