Splitting in Source–Terminal Network Reliability Estimation

Héctor Cancela
Universidad de la República, Montevideo, Uruguay

Leslie Murray
Universidad Nacional de Rosario, Rosario, Argentina

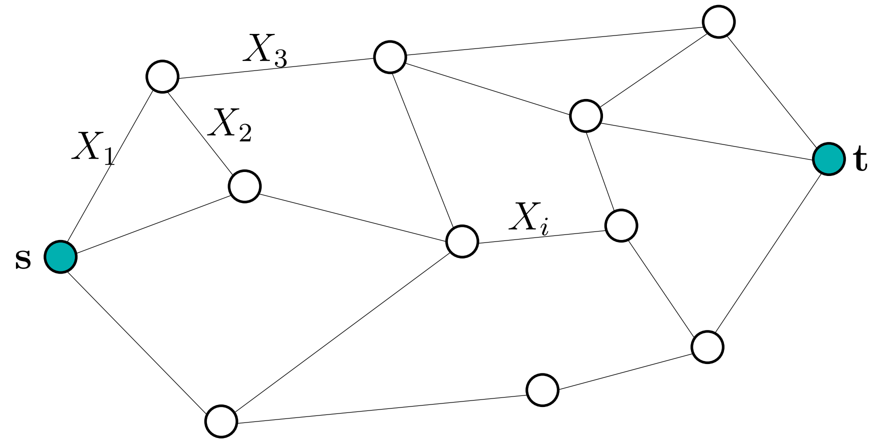
Gerardo Rubino
IRISA/INRIA, Rennes, France

RESIM
7th International Workshop on Rare Event Simulation
September 24-26 2008, Rennes, France

Talk Outline

- Network reliability model.
- Splitting method.
- Experimental results.
- Conclusions and future work.

Source-terminal network reliability model



$$X_i = \begin{cases} 1 \;\rightarrow\; \text{link } i \textbf{ operational} & \mathbb{P}[X_i = 1] = r_i \\ 0 \;\rightarrow\; \text{link } i \text{ link } \textbf{failed} & \mathbb{P}[X_i = 0] = q_i = 1 - r_i \end{cases}$$

Links state vector: $\mathbf{X} = (X_1, X_2, \ldots, X_m)$

$$\Phi(\mathbf{X}) = \begin{cases} 1 \;\rightarrow\; \textbf{operational} \text{ network (i.e., } s,t\text{-connected).} \\ 0 \;\rightarrow\; \textbf{failed} \text{ network (i.e., } s,t\text{-unconnected)} \end{cases}$$

$$\begin{cases} R(\mathcal{G}) = \mathbb{P}[\text{network is } s,t\text{-connected}] = \mathbb{E}\{\Phi(\mathbf{X})\} \\ Q(\mathcal{G}) = \mathbb{P}[\text{network is not } s,t\text{-connected}] = \mathbb{E}\{1 - \Phi(\mathbf{X})\} \end{cases}$$

Construction process model

All links considered failed at time 0; τ_i time to repair, exponential r.v.

$$\mathbb{P}[\tau_i \le t] = 1 - e^{-\lambda_i t}$$

$X_i(t)$ state of link i at time t;

$$X_i(t) = \begin{cases} 0 \text{ if } t < \tau_i \rightarrow & \text{link } i \textbf{ failed at time } t \\ 1 \text{ si } t \ge \tau_i \rightarrow & \text{link } i \textbf{ operational at time } t \end{cases}$$

Link state vector: $\mathbf{X}(t) = (X_1(t), X_2(t), \ldots, X_m(t))$

If $\lambda_i = -\log(q_i) \rightarrow \mathbb{P}[X_i(1) = 1] = \mathbb{P}[\tau_i \le 1] = 1 - e^{\log(q_i)} = r_i$

If $\mathbb{P}[X_i(1) = 1] = r_i \rightarrow \begin{cases} R = \mathbb{E}\{\Phi(\mathbf{X}(1))\} \\ Q = \mathbb{E}\{1 - \Phi(\mathbf{X}(1))\} \end{cases}$

Standard Monte Carlo over Construction process

- $\mathbf{X}^{(j)}(t)$: iid samples from $\mathbf{X}(t)$ defined by $\{\tau_1, \tau_2, \ldots, \tau_m\}$

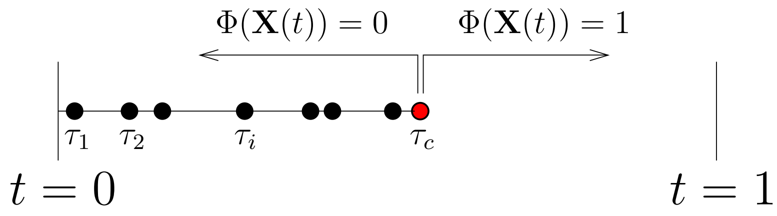
$$\widehat{R} = \frac{1}{N} \sum_{j=1}^{N} \Phi(\mathbf{X}^{(j)}(1)) \qquad \widehat{Q} = \frac{1}{N} \sum_{j=1}^{N} (1 - \Phi(\mathbf{X}^{(j)}(1)))$$

- Highly reliable network
  - "Many" $\tau_i < 1$
  - "Almost always" $\Phi(\mathbf{X}^{(j)}(1)) = 1$

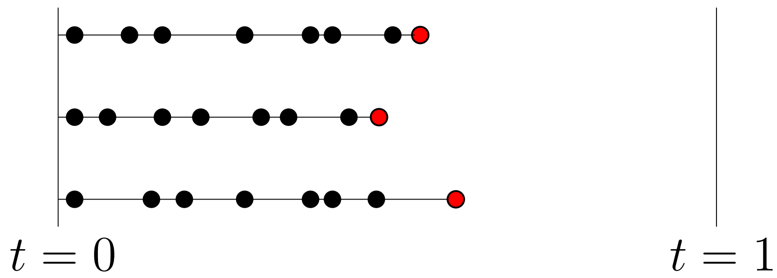
$$\Phi(\mathbf{X}^{(j)}(1)) = 0 \ \textbf{rare event}, \ \widehat{Q} \to 0$$

Relative error $RE = \dfrac{\mathbb{V}\{\widehat{Q}\}^{1/2}}{\mathbb{E}\{\widehat{Q}\}} = \left(\dfrac{1-Q}{NQ}\right)^{1/2} \approx \dfrac{1}{(NQ)^{1/2}} \longrightarrow \infty$

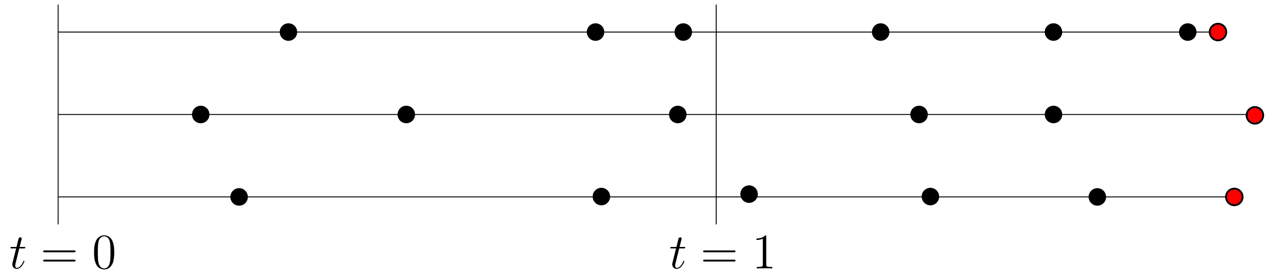
Network state evolution



- Highly reliable networks



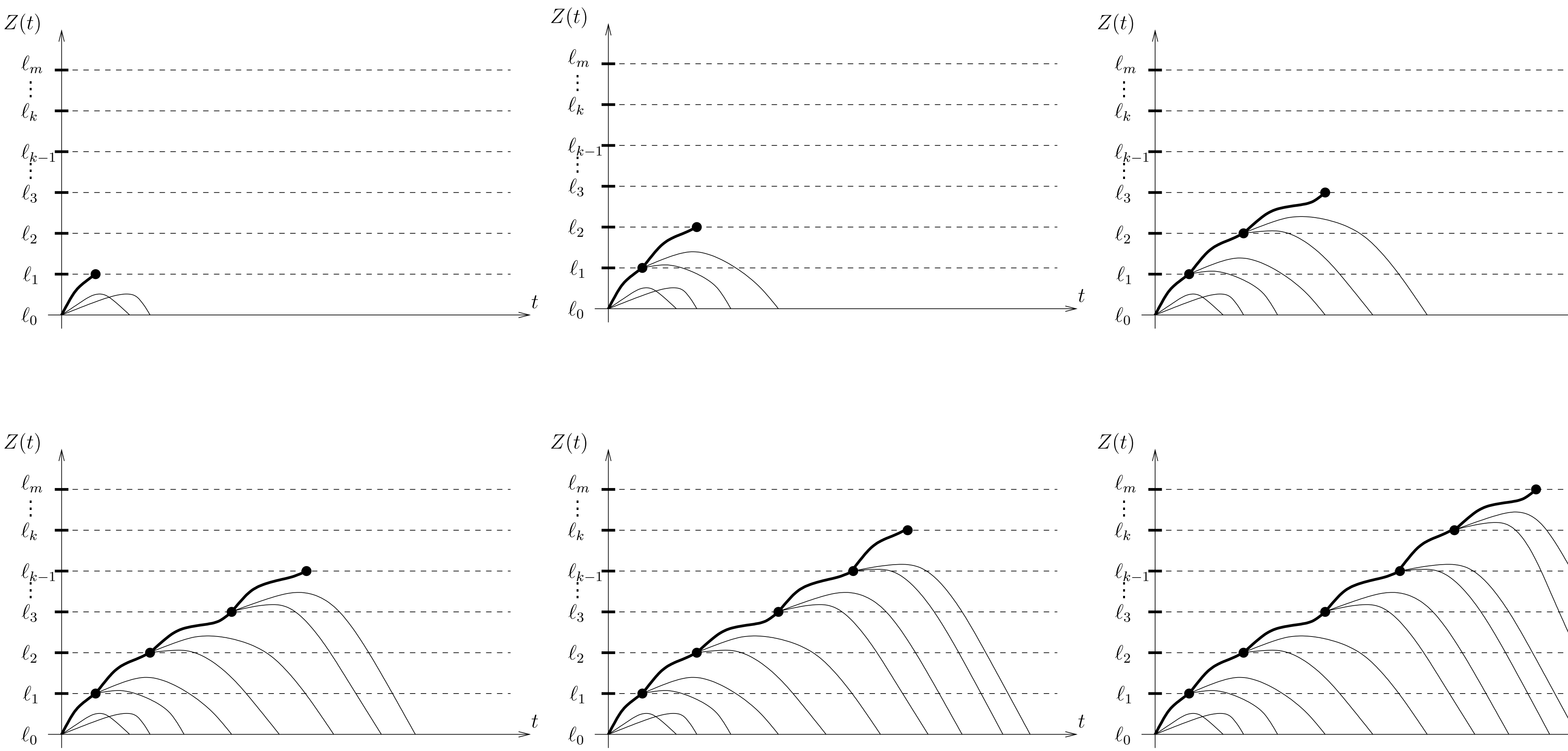
- Unreliable networks



Splitting–General description

- Well-known technique for rare event estimation in Markovian processes.
- Typical setting: given a Markov process Y and a function h, compute the (small) probability that Y enters a region of interest $A = \{y|h(y) \geq L\}$ before reaching another region $B\{y|h(y) \leq 0\}$.
- A series of intermediate regions are defined via thresholds $L_1, L_2, \ldots, L_n = L$. Whenever a trajectory reaches a threshold, it is "split" into a number of trajectories which might reach next threshold or "die".
- Estimator of measure of interest is given as product of the conditional estimations of reaching a threshold given that the previous one was reached.
- Technique used with good results in different contexts.

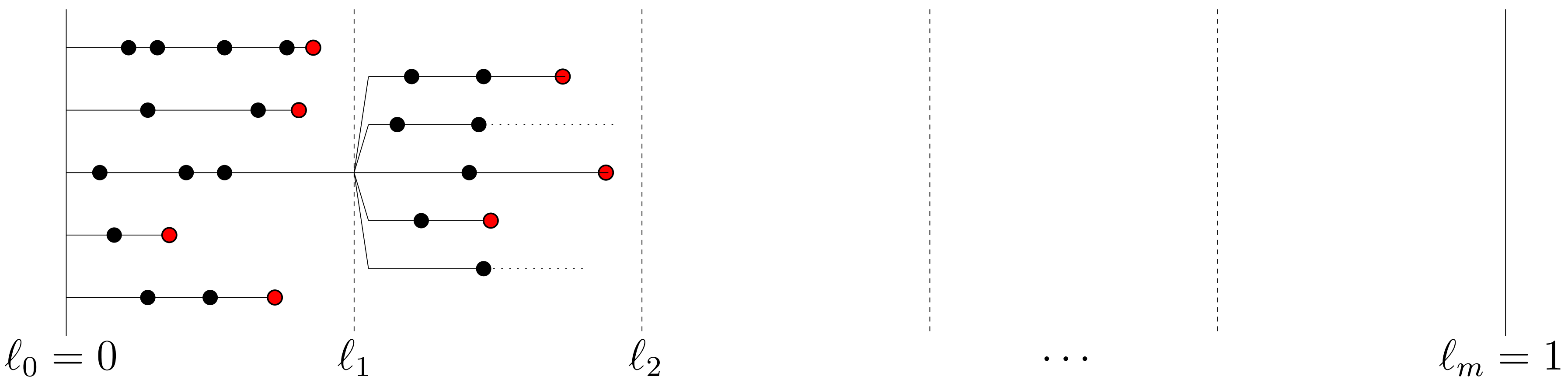
Splitting–graphical description



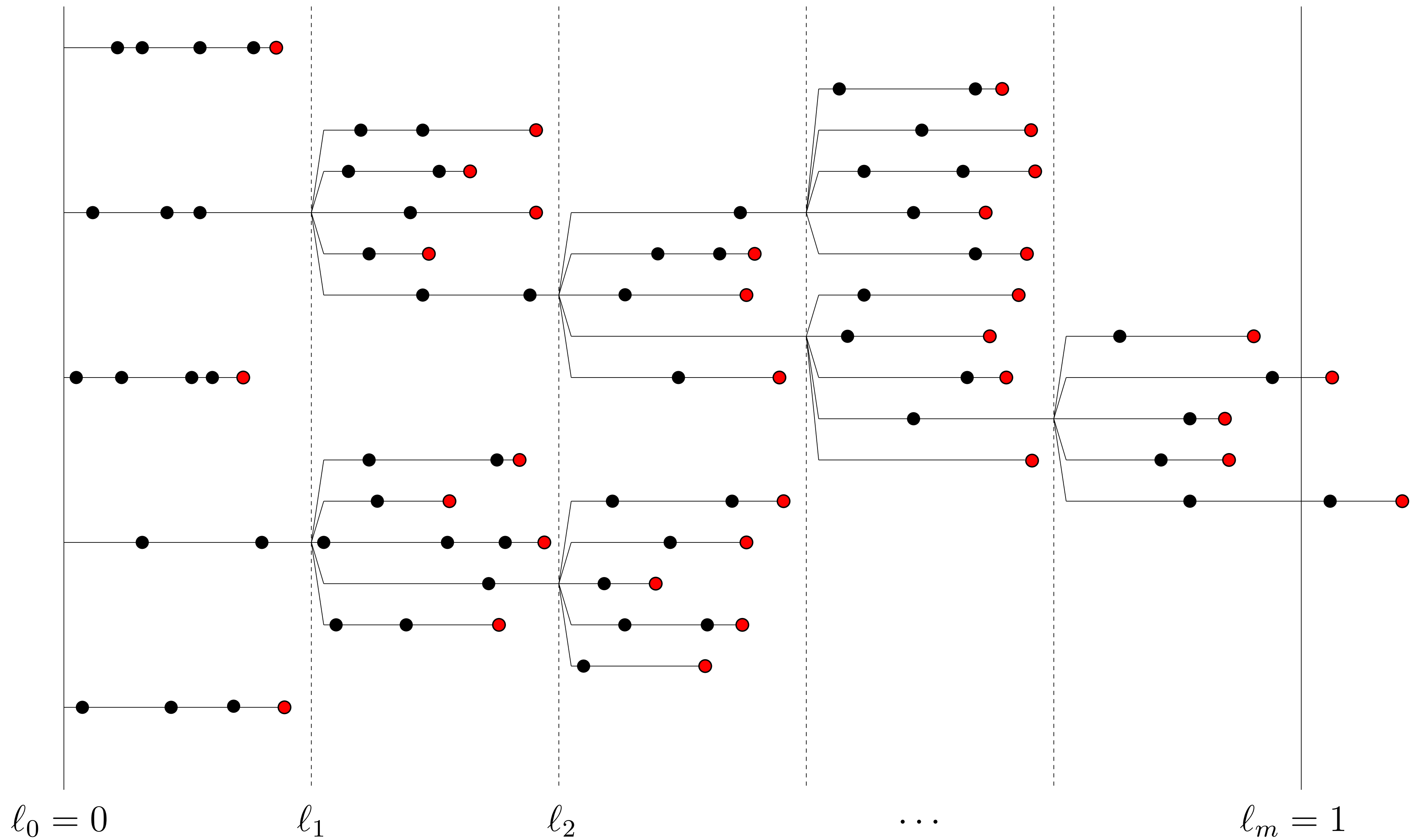
Splitting for network reliability (I)

- Network construction process: interesting trajectories are those such that $\Phi(\mathbf{X}(1)) = 0$. No natural h function for thresholds.
- Idea: to partition the trajectories based on a sequence of times $u_1, u_2, \ldots, u_n$.
- Then, a trajectory such that $\Phi(\mathbf{X}(u_k)) = 0$ will be cloned hoping that it will reach time u_{k+1} holding $\Phi(\mathbf{X}(u_{k+1})) = 0$; and a trajectory such that $\Phi(\mathbf{X}(u_k)) = 1$ will be killed.
- $Q = \mathbb{P}\{\mathbf{X}(1) = 0\}$ will be estimated as the product of the conditional probabilities over each threshold.

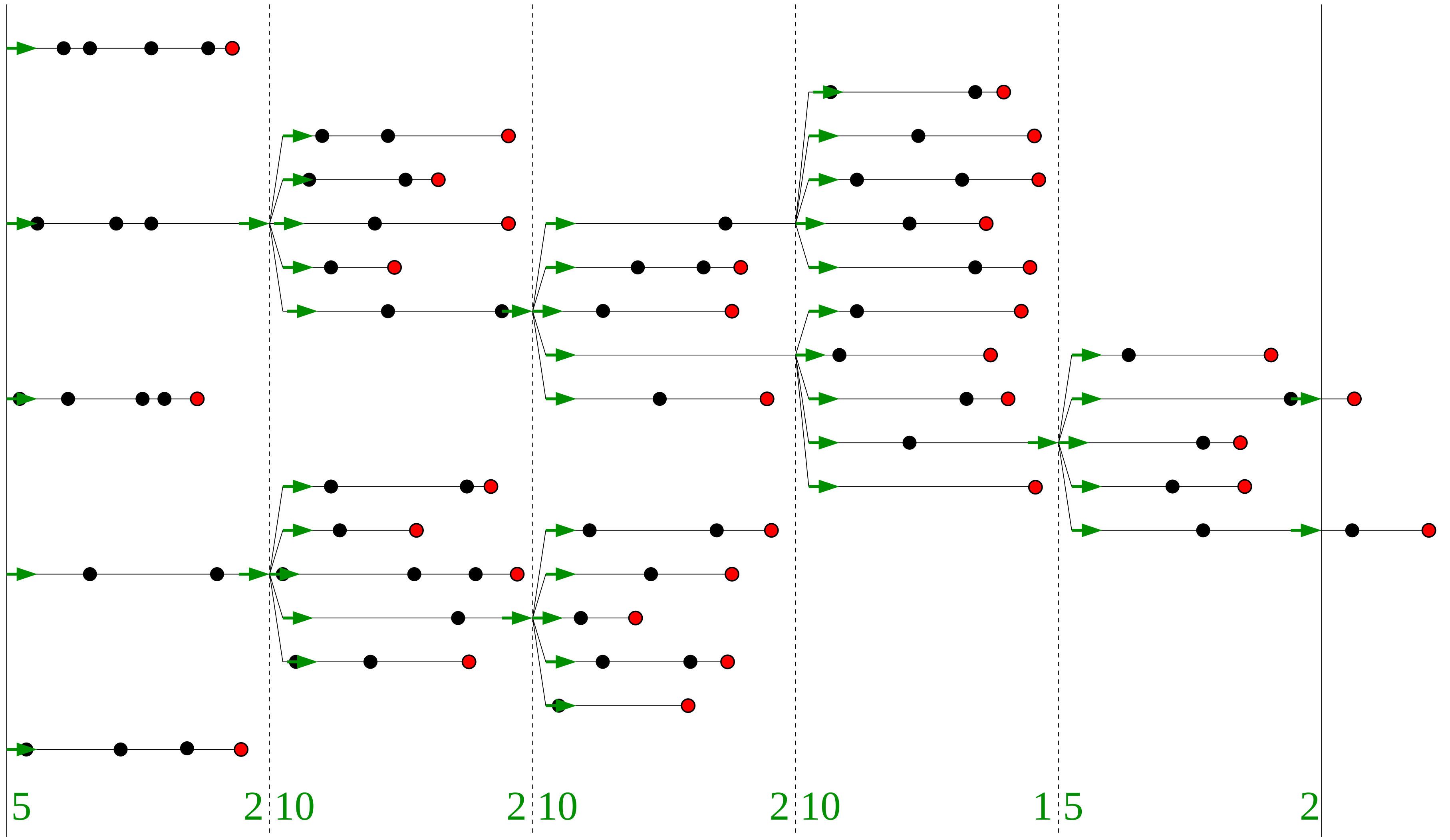
Splitting for network reliability (II)



Splitting for network reliability (III)



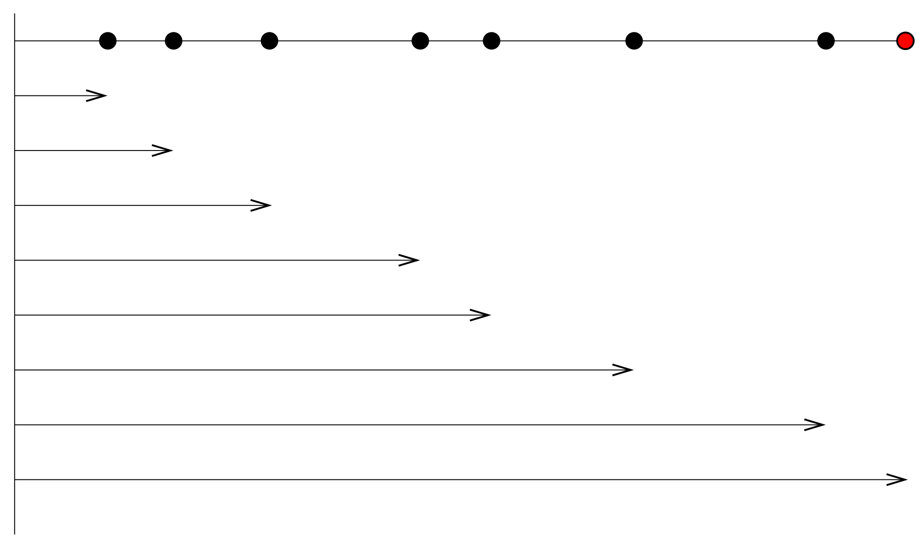
Splitting for network reliability (III)



$$\widehat{Q} = \frac{2}{5}\,\frac{2}{10}\,\frac{2}{10}\,\frac{1}{10}\,\frac{2}{5} \;=\; 6.4\times 10^{-4}$$

Sequence of up-times τ_i (1)

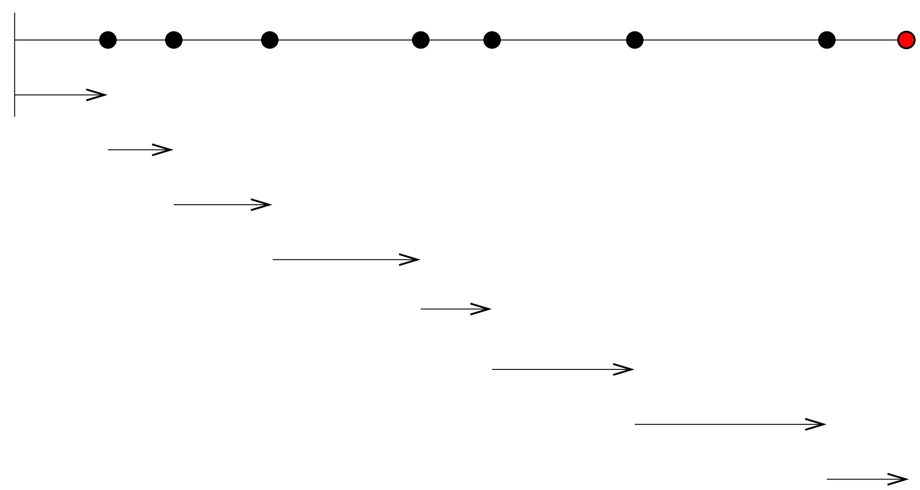
- Sample independently τ_i for every link i, exponential distribution with rate $\lambda_i = -\log(q_i)$;
- Sort all sampled τ_i in ascending order:



Sequence of up-times τ_i (2)

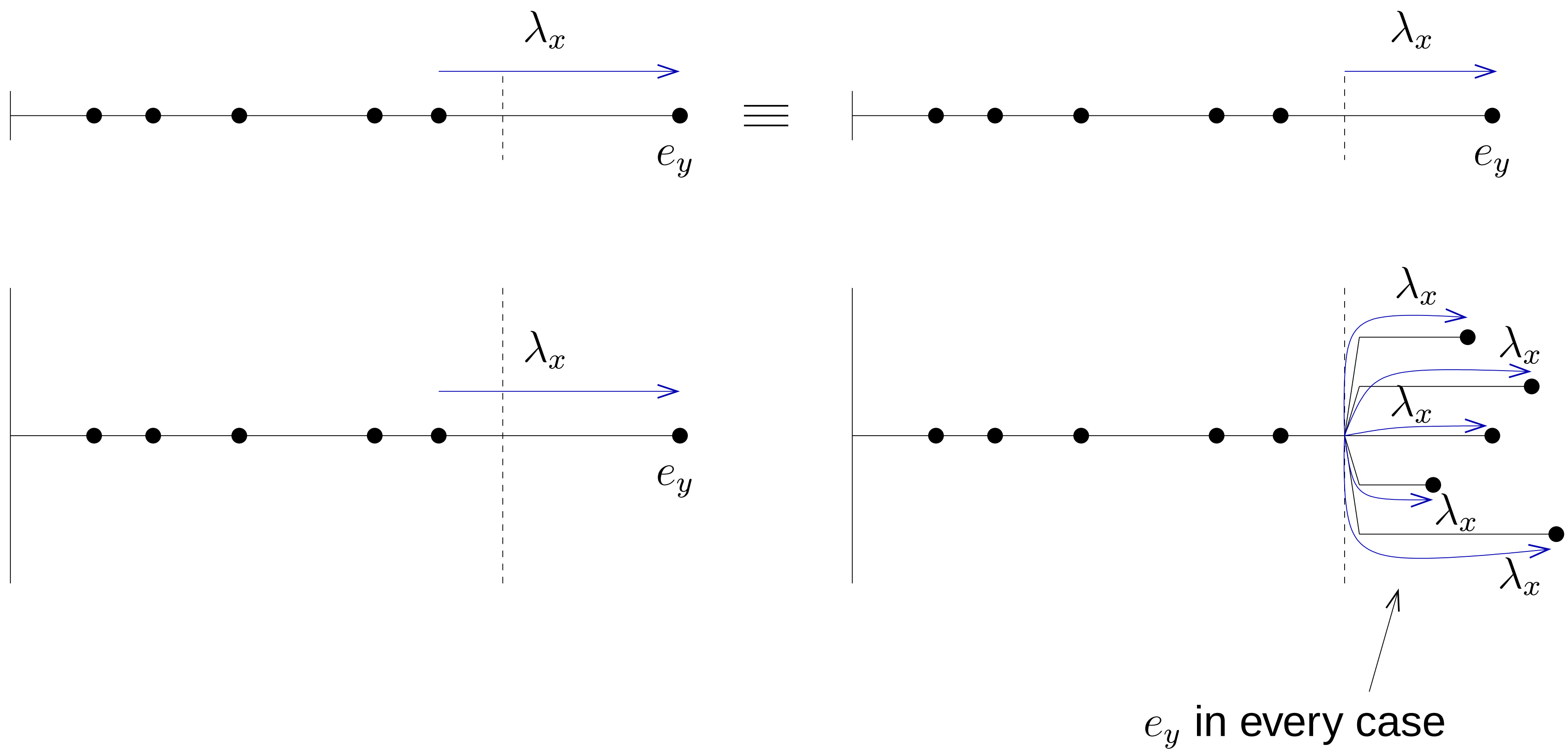
Let $P = \{$links not yet sampled$\}$

- Sample next link to go up from distribution $\mathbb{P}[e_i] = \frac{\lambda_i}{\sum_{e_j \in P} \lambda_j}$
- Sample elapsing time until link goes up from exponential distribution with rate $\sum_{e_j \in P} \lambda_j$



Splitting sequences

- Exponential distributions are memoryless:



Other implementation aspects

- How to determine number of thresholds?
  - Literature: minimize variance: $-(\log \widehat{Q})/2$
- How to determine the number of copies to make at each threshold?
  - Large enough to reach last threshold.
  - Not too large, else computational effort grows too quickly.
- Variants for splitting:
  - Fixed Splitting
  - Fixed Effort

Experimental setup

- 8 benchmark network topologies: dodecahedron, Arpanet (1972), complete graph C_{10}, bridge S_2 and graphs S_3, S_4, S_5, S_6.
- Equi-reliable links, reliability values `0.9`, `0.99`, `0.9999` and `0.999999`.
- The resulting source-terminal unreliabilities vary between `2.00e-02` and `2.00e-54`.
- Splitting implementation Fixed Effort.
- Number of trajectories at each threshold: 4000.
- Number of independent experiments: 200.

Summary of results

Red $(\mathcal{V}, \mathcal{E})$	$\widehat{Q}_{0.9}$	$t[seg]$	$RE[\%]$	$\widehat{Q}_{0.999999}$	$t[seg]$	$RE[\%]$
Dod $(20, 30)$	2.87e-03	173.47	**0.31**	2.03e-18	1,278.89	**0.57**
Arpanet $(21, 26)$	9.53e-02	110.39	**0.14**	6.00e-12	708.48	**0.44**
$K10$ $(10, 45)$	2.00e-09	802.65	**0.49**	2.01e-54	6,174.49	**1.19**
$S2$ $(4, 5)$	2.16e-02	13.86	**0.20**	2.01e-12	134.55	**0.38**
$S3$ $(8, 13)$	3.78e-03	56.06	**0.27**	2.99e-18	524.17	**0.50**
$S4$ $(14, 25)$	6.02e-04	163.70	**0.34**	4.03e-24	1,379.40	**0.71**
$S5$ $(22, 41)$	9.15e-05	372.12	**0.38**	4.98e-30	2,941.92	**0.88**
$S6$ $(32, 61)$	4.31e-05	834.86	**0.44**	5.38e-36	6,193.18	**2.61**

Other experiments

- Number of thresholds: $-(\log \widehat{Q})/2$ good results, near best values (always obtained by a slightly higher number of thresholds).
- Influence of number of trajectories vs. number of replications: number of trajectories must be large enough to guarantee reaching last threshold (4000); number of replications must be large enough to guarantee good variance estimation (100 or more).
- Comparison to Permutation Monte Carlo (another Construction Process based method) shows that, except for very small networks, Splitting attains better speedup values (for example, for S_5 and link reliabilities `0.9` up to `0.999999`, Splitting attains the same precision with an effort from 6 up to 28 times lower; for S_6, up to 473 times lower.

Conclusions

- Splitting adapted and applied in a new context.
- Performance of the method very robust in regard to network reliability values.
- Huge efficiency gains over Standard Monte Carlo.
- Good efficiency gains over Permutation Monte Carlo, specially for larger and more reliable networks.
- Future work: improve understanding of relation between number of thresholds and link reliability.
- Compare with other variance reduction methods.

Questions?