Splitting-Based Method for Network Reliability Estimation

Héctor Cancela, Leslie Murray, Gerardo Rubino

Splitting–Based Method for Network Reliability Estimation

Héctor Cancela [1] Leslie Murray [2] Gerardo Rubino [3]

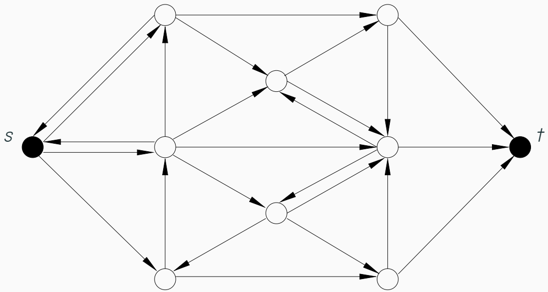
[1]Facultad de Ingeniería, UDELAR, Montevideo, Uruguay
cancela@fing.edu.uy

[2]Facultad de Ciencias Exactas, Ingeniería y Agrimensura, UNR, Rosario, Argentina
lmurray@fceia.unr.edu.ar

[3]INRIA–Rennes, Rennes, France
Gerardo.Rubino@inria.fr

The 12th International Conference on Monte Carlo Methods and Applications
Sydney, Australia

July, 2019

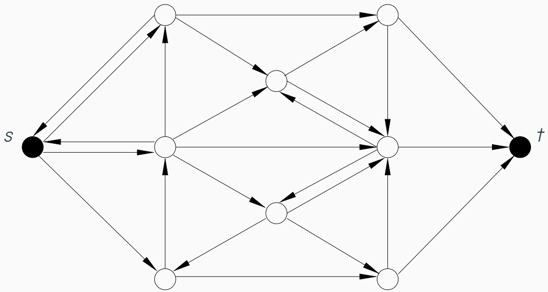


$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \begin{cases} \mathcal{V} : \text{set of } n \text{ nodes} \\ \mathcal{E} : \text{set of } m \text{ links} \\ \mathbf{X} : \text{capacity per link, } (X_1, \ldots, X_m) \end{cases}$$

Flow $\rightarrow$ | s: IN<OUT | intermediate: IN=OUT | t: IN>OUT

$V(\mathbf{X})$ = maximum amount of flow generated in s that can reach t

Stochastic Flow Network



- The capacities X_i, $i = 1, \ldots, m$, change due to failures.
- $\mathbf{X} = (X_1, \ldots, X_m)$, is a random vector in $\Omega = (\Omega_1, \ldots, \Omega_m)$.
- For a given demand, d, the **network unreliability** is: $\zeta = \mathbb{P}\{V(\mathbf{X}) < d\}$.
- If the network is highly reliable, $\zeta \ll 1$.

Aim of this talk

To introduce an efficient method for the estimation of ζ, when $\zeta \ll 1$, that allows a direct extension to the case of dependent components.

- The case of dependent components can be handled by means of the Marshall-Olkin copula.
- The rare event situation will be handled here using Splitting.
- In previous work, we used Splitting over the so-called Creation Process, introduced by Lomonosov et al., but this option is not directly adaptable to the case of dependent components.
- In this work–in–progress presentation, we will describe how to obtain low variance Splitting–based estimators using the Destruction Process instead.
- We will actually describe a *n* extension of the Destruction Process to the case of multi–valued structure functions (instead of the previously considered binary ones) and then, how to implement a Splitting–based procedure.

- In the most general discrete model:

$$\Omega_i = (0, M_{i_1}, M_{i_2}, \ldots, M_{i_{n_i}}), \quad i = 1, \ldots, m$$

where: $0 < M_{i_1} < M_{i_2} < \ldots < M_{i_{n_i}}$.

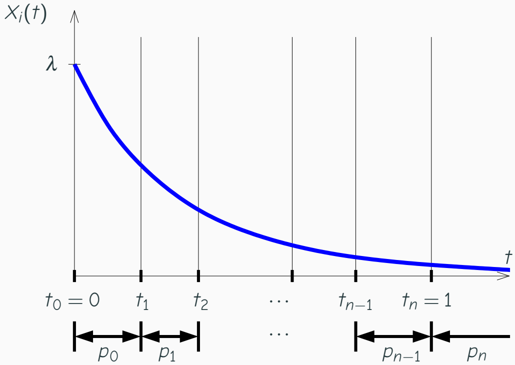
- In the model to be introduced:

$$\Omega_i = (0, M_1, M_2, \ldots, M_n)$$

where: $0 < M_1 < M_2 < \ldots < M_n$.

  - If $X_i = M_n$, link i is fully operational.
  - If $0 < X_i < M_n$, link i is partially failed.
  - If $X_i = 0$, link i is completely failed.

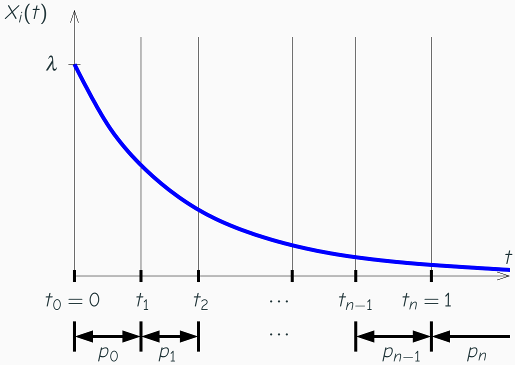
$$X_i = \begin{cases} M_n & \text{w.p. } p_n, \\ & \vdots \\ M_1 & \text{w.p. } p_1, \\ 0 & \text{w.p. } p_0. \end{cases}$$



- Let $\tau_i \sim \text{Exp}(\lambda)$ be the failure time of link i.

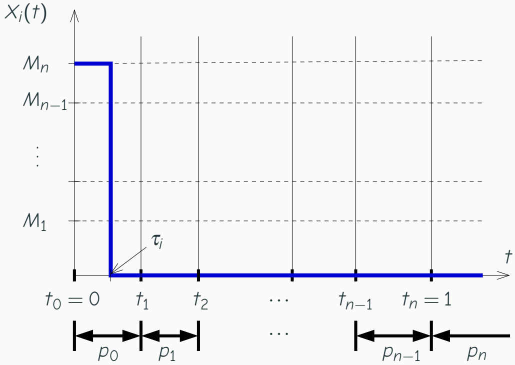
- Let $\{\lambda, t_1, t_2, \ldots, t_{n-1}\}$ be a set such that $\rightarrow$

$$\begin{cases} 0 < \tau_i \leq t_1 & \text{w.p. } p_0, \\ t_1 < \tau_i \leq t_2 & \text{w.p. } p_1, \\ \vdots & \\ t_{n-1} < \tau_i \leq 1 & \text{w.p. } p_{n-1}, \\ 1 < \tau_i & \text{w.p. } p_n. \end{cases}$$



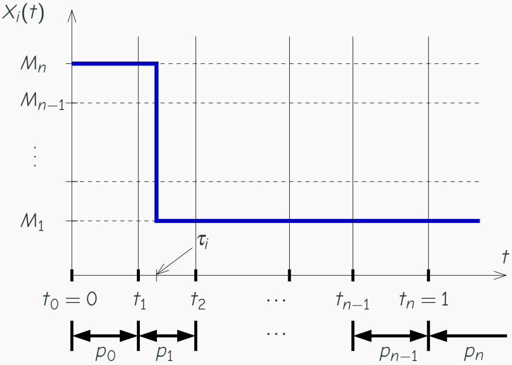
It is simple to prove that:

$$\begin{cases} \lambda = -\ln(p_n), & \\ t_j = \dfrac{\ln(1 - p_0 - p_1 - \ldots - p_{j-1})}{\ln(p_n)} & j = 1, \ldots, n-1. \end{cases}$$



Let $X_i(t) = M_n,\ 0 \leq t < \tau_i$,

- if $0 < \tau_i \leq t_1$ set $X_i(\tau_i) = 0$ and keep this value forever,
  if $t_1 < \tau_i \leq t_2$ set $X_i(\tau_i) = M_1$ and keep this value forever,
  $\vdots$
  if $t_{n-1} < \tau_i \leq 1$ set $X_i(\tau_i) = M_{n-1}$ and keep this value forever,
  if $1 < \tau_i$ leave $X_i(t) = M_n$ forever.

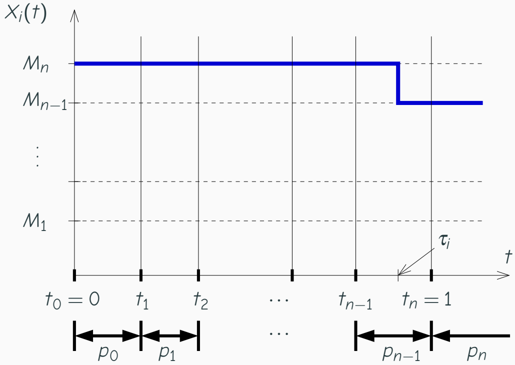


Let $X_i(t) = M_n,\ 0 \le t < \tau_i$,

- if $0 < \tau_i \le t_1$ set $X_i(\tau_i) = 0$ and keep this value forever,
- if $t_1 < \tau_i \le t_2$ set $X_i(\tau_i) = M_1$ and keep this value forever,

$\vdots$

- if $t_{n-1} < \tau_i \le 1$ set $X_i(\tau_i) = M_{n-1}$ and keep this value forever,
- if $1 < \tau_i$ leave $X_i(t) = M_n$ forever.



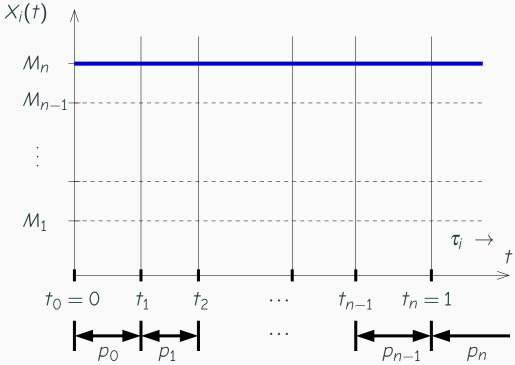
Let $X_i(t) = M_n,\ 0 \le t < \tau_i$,

- if $0 < \tau_i \le t_1$ set $X_i(\tau_i) = 0$ and keep this value forever,
- if $t_1 < \tau_i \le t_2$ set $X_i(\tau_i) = M_1$ and keep this value forever,

$\vdots$

- ▶ if $t_{n-1} < \tau_i \le 1$ set $X_i(\tau_i) = M_{n-1}$ and keep this value forever,
- if $1 < \tau_i$ leave $X_i(t) = M_n$ forever.

Multi–Level Destruction Process – Evolution



Let $X_i(t) = M_n,\ 0 \leq t < \tau_i$,

if	$0 < \tau_i \leq t_1$	set	$X_i(\tau_i) = 0$ and keep this value forever,
if	$t_1 < \tau_i \leq t_2$	set	$X_i(\tau_i) = M_1$ and keep this value forever,
	$\vdots$		
if	$t_{n-1} < \tau_i \leq 1$	set	$X_i(\tau_i) = M_{n-1}$ and keep this value forever,
▶ if	$1 < \tau_i$	leave	$X_i(t) = M_n$ forever.

Finally:

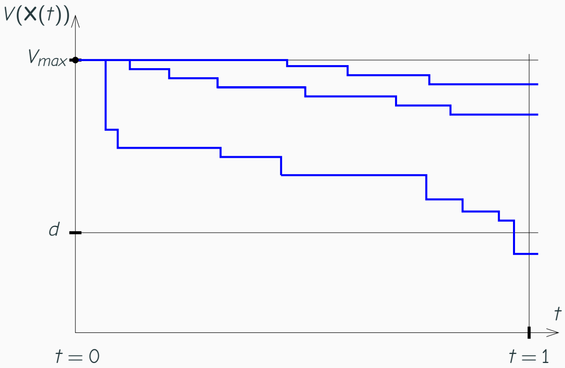
$$X_i(1) = \begin{cases} M_n & \text{w.p. } p_n, \\ & \vdots \\ M_1 & \text{w.p. } p_1, \\ 0 & \text{w.p. } p_0. \end{cases}$$

Thus, observing this dynamic model at $t = 1$ is the same as observing the static model.

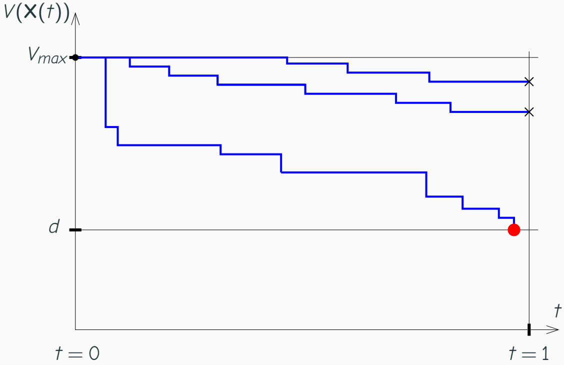
The Multi–Level Destruction Process is a generalization of the well-known *Destruction Process*, designed for static network models in which links can only be operational or failed.

How can the the Multi–Level Destruction Process be used for estimating the reliability ζ by simulation?

Crude Monte Carlo over the Multi-Level Destruction Process

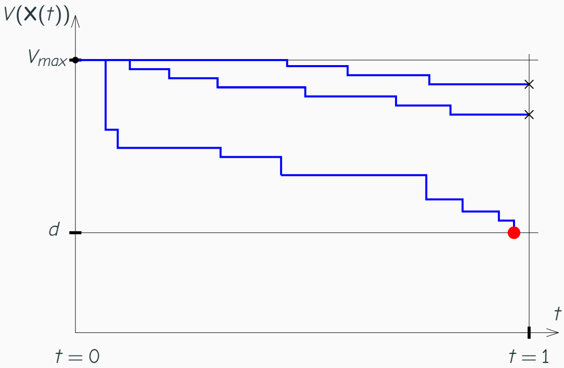


- V_{max} is the maximum possible value of $V(\mathbf{X}(t))$ (all links at maximum capacity).
- According to the links' failures, $V(\mathbf{X}(t))$ becomes a constant piece-wise decreasing function of time.
- It only matters if, at $t = 1$, the trajectories are above or below the demand, d.

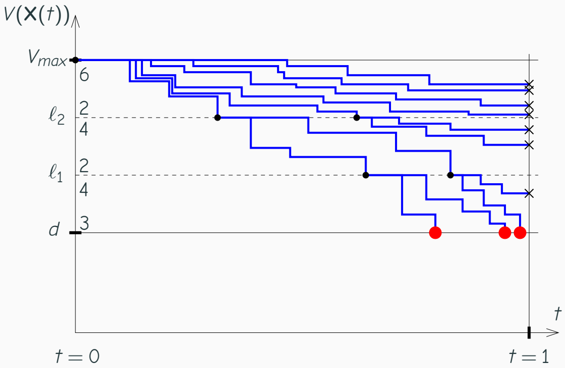


- The trajectories that down–cross d before $t = 1$ will be below d in $t = 1$.
- Simulation is not worth continuing from the red points.
- The crude Monte Carlo estimation of ζ is the ratio between the number of red points and the number of trajectories started at $t = 0$.

$$\widehat{\zeta} = \frac{1}{3} = 0.333$$

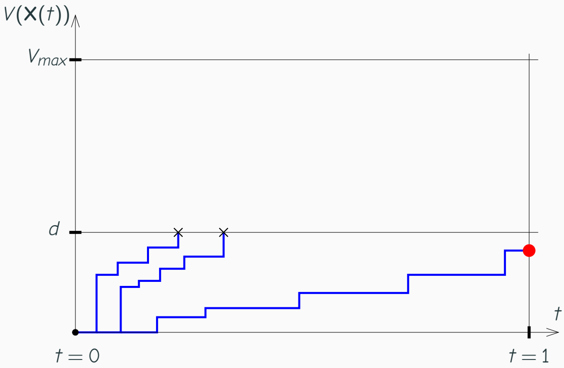


- For highly reliable networks, trajectories will rarely "fall down".
- There will be very few red points.
- In order to make estimates more efficient, trajectories should be artificially forced "downwards".
- This goal is achieved by means of the following application of Splitting.

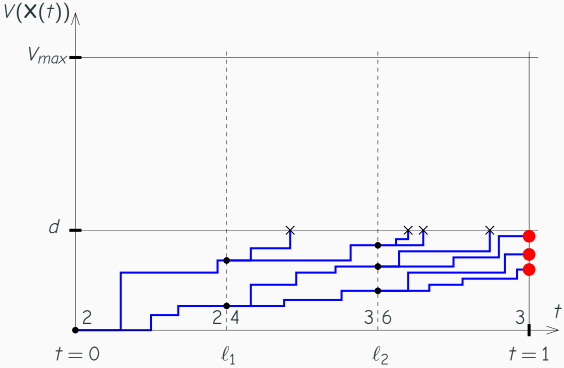


- Set the thresholds $\ell_1, \ell_2, \ldots,$ on the $V(\mathbf{X}(t))$ axis.
- Start, at every crossing point, multiple copies of those trajectories that have down–crossed the corresponding threshold before $t = 1$.
- The Splitting estimation for this example is: $\widehat{\zeta} = \frac{2}{6}\,\frac{2}{4}\,\frac{3}{4} = 0.125$

With the same purpose, Splitting has been applied over the Creation Process. Which are the differences and the benefits of one option over the other?



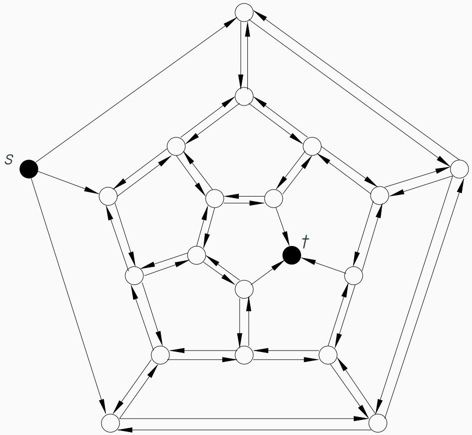
- The crude Monte Carlo estimation of ζ is the ratio between the number of red points and the number of trajectories started at $t = 0$.
- For highly reliable networks, trajectories will rarely "go to the right".
- In order to make estimates more efficient, trajectories should be artificially forced "to the right".
- This goal is achieved by means of the following application of Splitting.



- Thresholds $\ell_1, \ell_2, \dots,$ are set on the t axis.
- At every crossing point, multiple copies of those trajectories that have right–crossed the corresponding threshold below d, are launched.
- The Splitting estimation for this example is: $1 - \widehat{\zeta} = \frac{2}{2}\ \frac{3}{4}\ \frac{3}{6} = 0.375.$

Both methods are efficient, however the Destruction Process, in which links fail (rather than being repaired), fits more naturally than the Creation Process to the Marshall–Olkin copula, with dependent link failures.

A Numerical Illustration



$$X_i = \begin{cases} 8 & \text{w.p. } 0.9899, \\ 4 & \text{w.p. } 0.0100, \\ 0 & \text{w.p. } 0.0001. \end{cases} \quad \rightarrow \quad V_{max} = 24$$

A Numerical Illustration

$\widehat{\zeta}$	RE	Th	N	t	d
1.22 E−07	2.82%	20	10^6	54	12
1.19 E−07	2.27%	21	10^6	96	12
1.19 E−07	2.01%	22	10^6	149	12
1.19 E−07	2.01%	23	10^6	201	12
1.19 E−07	1.87%	24	10^6	371	12
6.05 E−10	4.05%	26	10^6	58	8
6.06 E−10	4.05%	27	10^6	62	8
5.63 E−10	2.95%	28	10^6	112	8
5.83 E−10	2.59%	29	10^6	152	8
6.27 E−10	2.30%	30	10^6	260	8

Where:

- $\widehat{\zeta}$ Unreliability estimation.
- RE Relative error: $\mathrm{SD}(\widehat{\zeta})/\mathbb{E}(\widehat{\zeta})$
- Th Number of thresholds.
- N Number of Monte Carlo trials.
- t Simulation time in seconds.
- d Network demand.

Concluding Remarks

- Preliminary tests show that Splitting over the Multi–Level Destruction Process is a very efficient method for estimating the unreliability of highly reliable flow networks.
- Some other known (published) Splitting–based methods work on the Creation Process, rather than the Destruction Process.
- For methods that work on the Creation Process, thresholds are set on the t axis.
- To set the thresholds on the t axis in methods based on the Destruction Process yields no significant improvement.
- The main and most interesting feature of the proposed method is that the Destruction Process fits naturally to the Marshall–Olkin copula network model with dependent links' failures.
- We are currently exploring this approach with models in which there are correlations between the behavior of their components.

Main References

Zdravko I. Botev et al. “Static Network Reliability Estimation Under the Marshall-Olkin Copula”. In: *ACM Trans. Model. Comput. Simul.* 26.2 (Jan. 2016), 14:1–14:28. ISSN: 1049-3301. DOI: 10.1145/2775106. URL: http://doi.acm.org/10.1145/2775106.

H. Cancela, L. Murray, and G. Rubino. “Efficient Estimation of Stochastic Flow Network Reliability”. In: *IEEE Transactions on Reliability* (2019), pp. 1–17. ISSN: 0018-9529. DOI: 10.1109/TR.2019.2897322.