Stochastic flow networks, rare events, dependent components and Splitting Monte Carlo techniques

Gerardo Rubino

Stochastic flow networks, rare events, dependent components and Splitting Monte Carlo techniques

Gerardo Rubino, INRIA, France

Joint work with
Leslie Murray, Faculty of Exact Sciences, U. of Rosario, Argentina,
and Héctor Cancela, Faculty of Engineering, UDELAR, Uruguay

May 25, 2022

Informal context

- This work is about analyzing dependability properties of complex systems, by means of Monte Carlo techniques.
- More specifically, we consider transportation systems where some fluid is sent through a network from a source to a destination, going from node to node through directional links having some *capacity*.
- The flow can make a fork at a given node, or a join.
- The links capacities are random variables, and we call failures the events "moving from the standard (nominal) maximum capacity to a smaller value". They are usually supposed to be independent of each other. They are also called the model's *components*.
- The problem is a static one (no time variable). At a given time, where the system is considered, the links have some capacities, sampled from the capacities' distributions.
- Because of the links capacities, there is a maximal amount of flow that can be transported by the network, MF, from source to destination, which is a random variable.

Informal context

- We are given a *demand* value, a minimal amount of fluid we need to be able to transport. We want to evaluate the number $\zeta = \mathbb{P}\{MF < d\}$, an important dependability metric in this context, an *unavailability* one.
- In many situations, and the one considered here, the event $\{MF < d\}$ is rare, that is, $\zeta \ll 1$. How to estimate it is the topic of this talk.
- The main families of techniques to deal with rare events are Importance Sampling and Splitting (also appearing under other different names), plus some other special ones such as Recursive Variance Reductions.
- In this paper we deal with Splitting, which is specifically designed for problems defined on stochastic processes (here, the setting is static).
- Because of that, we must first transform the static model into a dynamic one.
- After presenting the transformation, we will describe a particular implementation of the Splitting approach that gives very good results.
- Then, we will relax the independence assumption (between capacities), and show how the proposed method can also deal with this extension to the original model. The method was actually designed to deal with such an extension.
- To deal with the dependent components case, we use Marshall-Olkin copulas.

Outline

The talk will consist of the following points:

1 – a brief refresher on the Splitting method,

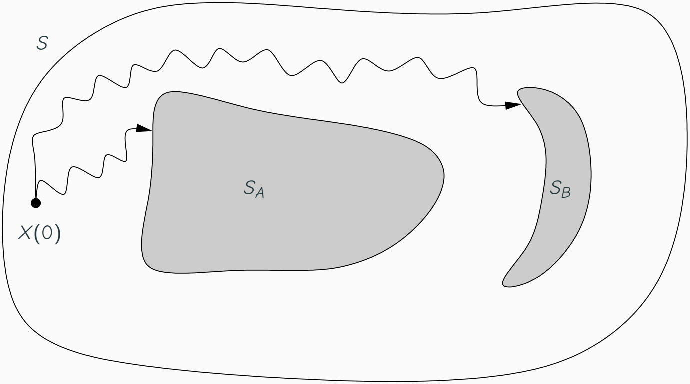
2 – the flow model,

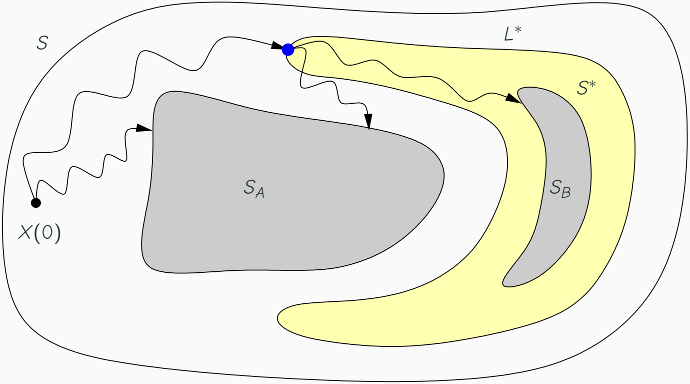
3 – the Creation and Destruction Processes (CP and DP)
and our multilevel extensions,

4 – our Splitting method, designed to work also with dependent components.

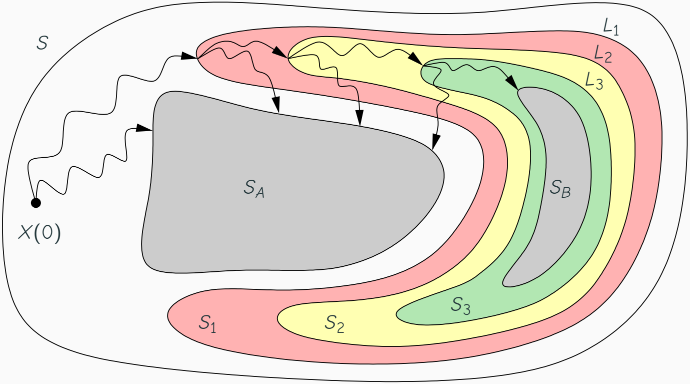
1—Splitting

- We have a stochastic process $X = \{X(t)\}_{t \geq 0}$ living in some space S, and two subsets of states S_A and S_B. Assume them disjoint, with $X(0) \notin S_A, S_B$.
- Let us denote τ_A (resp. τ_B) the hitting times of X in S_A (resp. in S_B).
- We are interesting in evaluating $\zeta = \mathbb{P}\{\tau_B < \tau_A\}$, and we consider the case where $\tau_B \ll \tau_A$ with high probability.



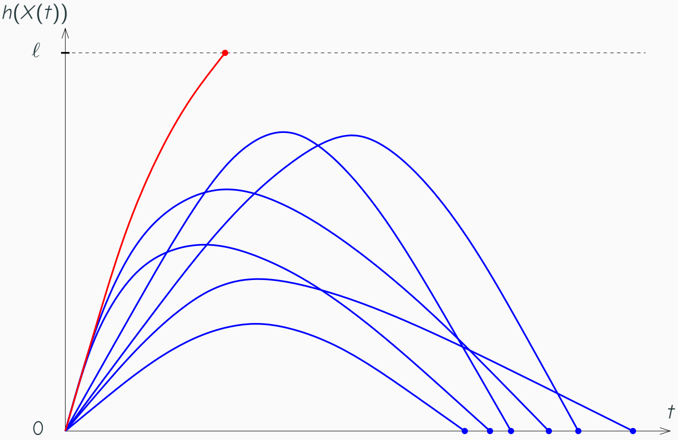


- First idea: simulate several independent copies of X. When one of them gets close to S_B at some time τ, make several copies of it, starting at τ.
- For that purpose, define an intermediate set $S^* \supset S_B$, with $S^* \cap S_A = \emptyset$, with border L^*, and split X when it touches L^*.

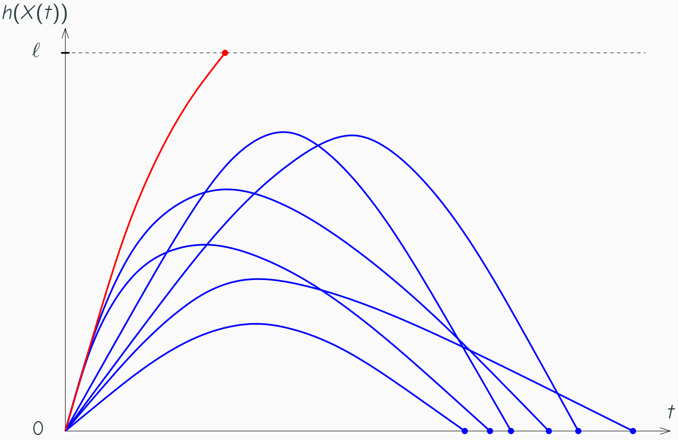


- The procedure can be applied recursively, cloning trajectories that cross the borders L_1,L_2,L_3 of a sequence of subspaces $S_1 \supset S_2 \supset S_3 \supset S_B$.
- Second (possible) idea: when a trajectory that crossed border L_i comes back at L_i before reaching L_{i+1}, kill it (the RESTART variation).

- This is widely used in queuing models for performance evaluation of systems, and in dependability models.
- In some cases we want to evaluate $\mathbb{P}(\tau_B < T)$ where T is an a.s. finite stopping time.
- In other cases, this type of probabilities help in analyzing other targets. Example:
  - suppose that the states in S_B are "bad" states, and that $\tau_A = \tau_0$, the return time to the initial state.
  - τ_B is the system's *life-time*.
  - We are interested in evaluating the MMTF of the system, the Mean Time To Failure, $= \mathbb{E}(\tau_B)$. In the highly reliable case, MTTF$\gg 1$.
  - It can be shown that
  $$\mathbb{E}(\tau_B) = \frac{\mathbb{E}\big(\min(\tau_0, \tau_B)\big)}{\mathbb{P}(\tau_B < \tau_0)}.$$
  The numerator is easy to estimate; the estimation of the denominator is a typical rare event problem.

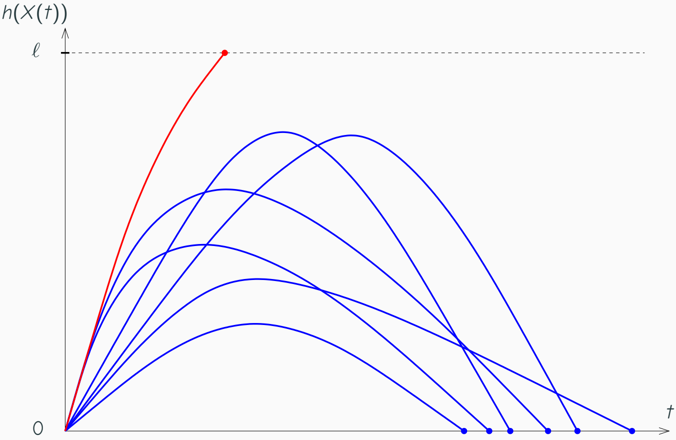


- Everything is better controlled using a real function $h : S \to \mathbb{R}$, called *importance function*, such that, for instance, $x \in S_A \iff h(x) \le 0$ and $x \in S_B \iff h(x) \ge \ell > 0$.



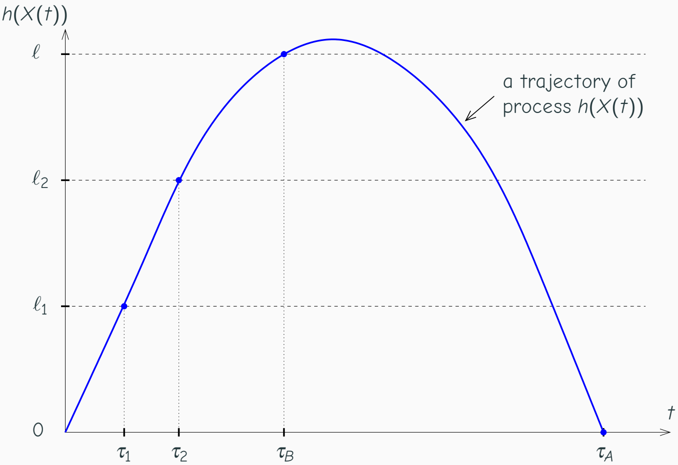
- In the models we are dealing with, $h(X(t))$ will tend to stay low, to come back quickly to 0 (or to negative values, depending on how we defined h), and it will be rare to observe it getting close to ℓ.

- In a queueing system modeled by some process, a typical example is $h(x) = \#$ of customers in some queue, when system's state is x.
- In a dependability model, a typical example is $h(x)$ = # of failed components, when system's state is x.
- In both cases, $h(x)$ measures somehow how close we are to the bad states (too many customers in the queue, or too many components down in the dependability case).

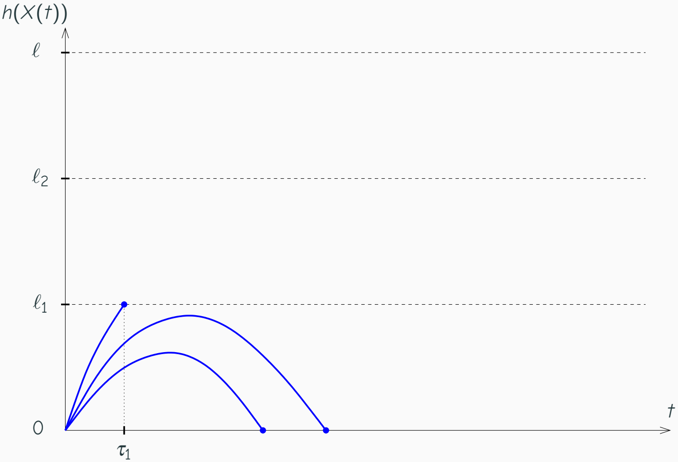


$$\zeta = \frac{1}{7}$$

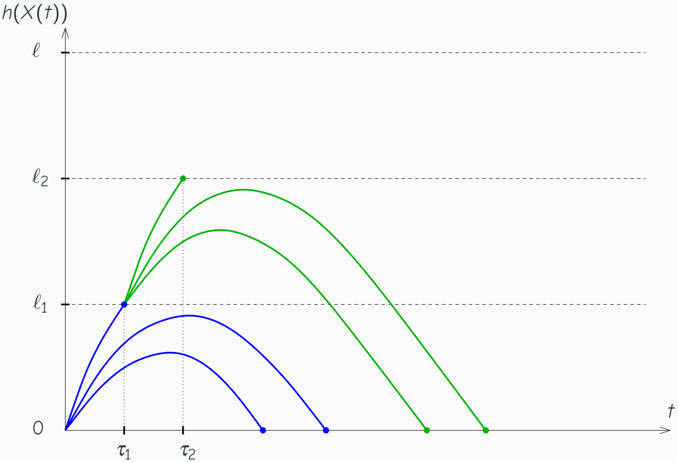
We estimate ζ by dividing the number of trajectories reaching ℓ by the total number of copies simulated.



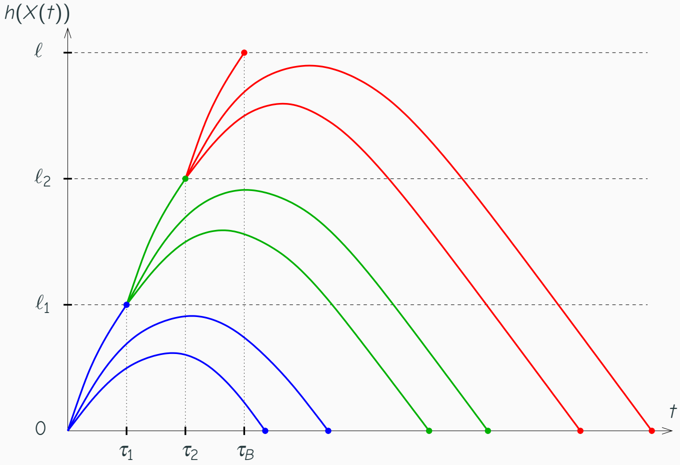
- Now, suppose that the set of values of h is partitioned using thresholds $\ell_1, \ell_2, \ldots,$ defining an associated sequence of embedded subspaces of S.
- In the figure, $x \in S_1 \iff h(x) \geq \ell_1$, $x \in S_2 \iff h(x) \geq \ell_2$, plus, for instance, $x \in S_B \iff h(x) \geq \ell$ and $x \in S_A \iff h(x) = 0$.



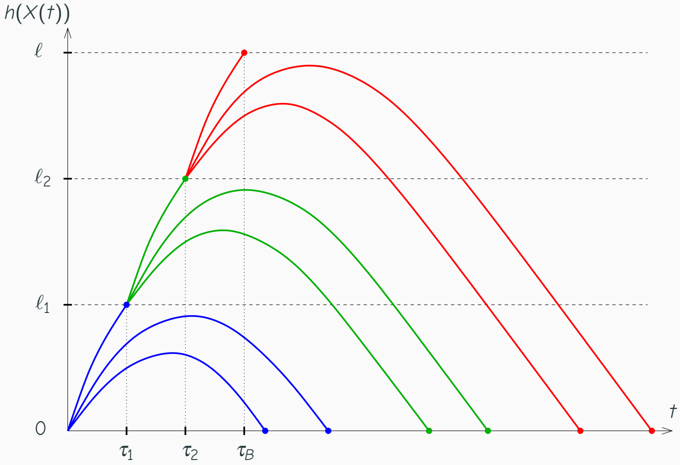
- Crossing the levels $\ell_1, \ell_2, \ldots$ is equivalent to crossing the borders $L_1, L_2, \ldots$ defined on S (in the previous description).
- Now, each time a trajectory reaches level or threshold ℓ_i (at time τ_i), we clone it.



- Crossing the levels $\ell_1, \ell_2, \ldots$ is equivalent to crossing the borders $L_1, L_2, \ldots$ defined on S (in the previous description).
- Now, each time a trajectory reaches level or threshold ℓ_i (at time τ_i), we clone it.



- Crossing the levels $\ell_1, \ell_2, \ldots$ is equivalent to crossing the borders $L_1, L_2, \ldots$ defined on S (in the previous description).
- Now, each time a trajectory reaches level or threshold ℓ_i (at time τ_i), we clone it.



- D_i = event $\{\tau_i < \tau_A\}$.
- If we have k thresholds, $D_k \subset D_{k-1} \subset \cdots \subset D_2 \subset D_1$

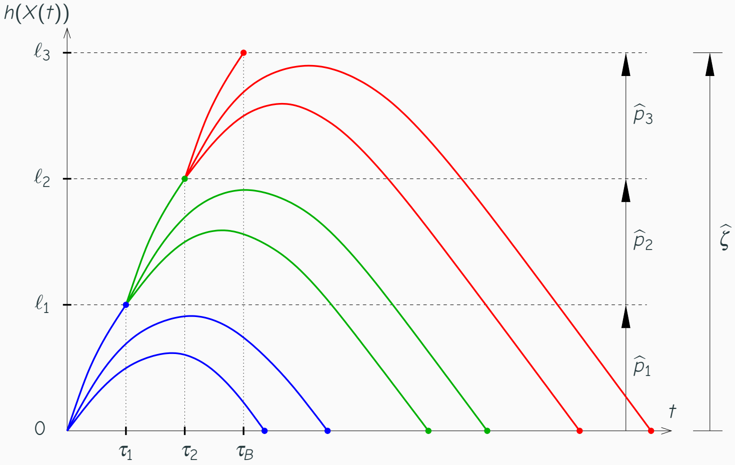
- Assume we have k thresholds $\ell_1, \ell_2, \ldots, \ell_k = \ell$. The event of interest is D_k.
- Easy to check that we have

$$\mathbb{P}\{D_k\} = \underbrace{\mathbb{P}\{D_k \,|\, D_{k-1}\}}_{p_k}\ \underbrace{\mathbb{P}\{D_{k-1} \,|\, D_{k-2}\}}_{p_{k-1}} \cdots \underbrace{\mathbb{P}\{D_2 \,|\, D_1\}}_{p_2}\ \underbrace{\mathbb{P}\{D_1\}}_{p_1} = \zeta = \prod_{h=1}^{k} p_h$$

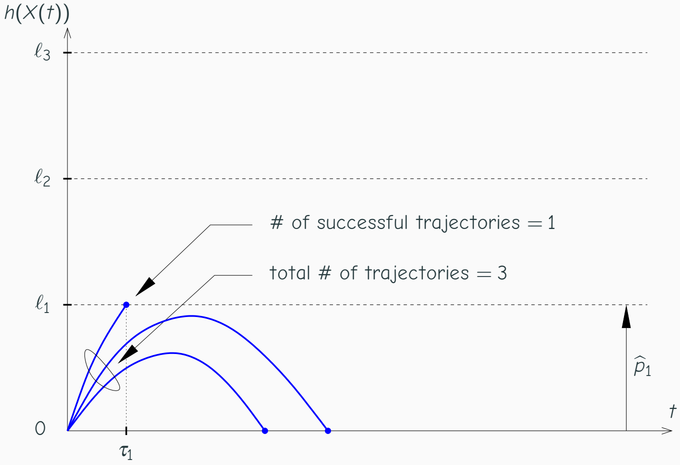
- At the heart of Splitting, we have the following result:

$$\widehat{\zeta} = \prod_{h=1}^{k} \widehat{p}_h \quad \rightarrow \quad \mathbb{E}\{\widehat{\zeta}\} = \zeta,$$

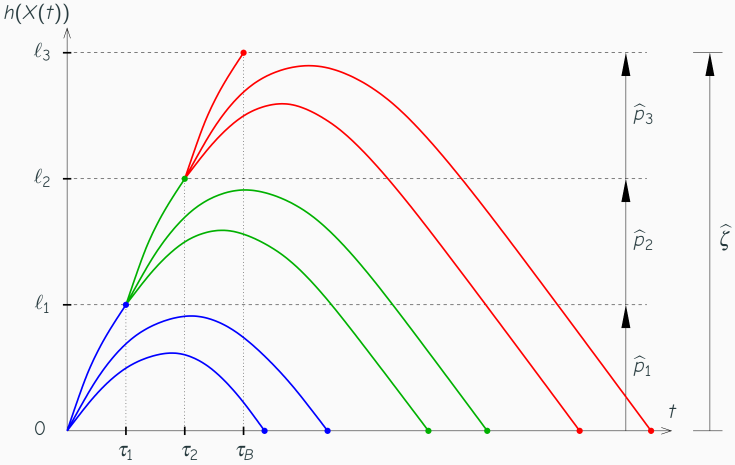
  where $\widehat{p}_h$ is the standard estimator of p_h.
- That is, we have an unbiased estimator of the target built from unbiased (crude) estimators associated with a single threshold case (which is built such that reaching the threshold is not rare).



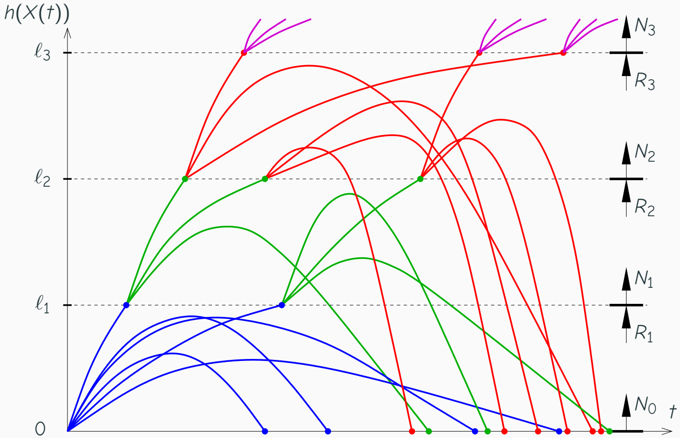
$$\widehat{\zeta} = \widehat{p}_1 \widehat{p}_2 \widehat{p}_3$$



$$\widehat{p}_1 = \frac{1}{3}$$



$$\widehat{\zeta} = \frac{1}{3}\,\frac{1}{3}\,\frac{1}{3} = \frac{1}{9}$$



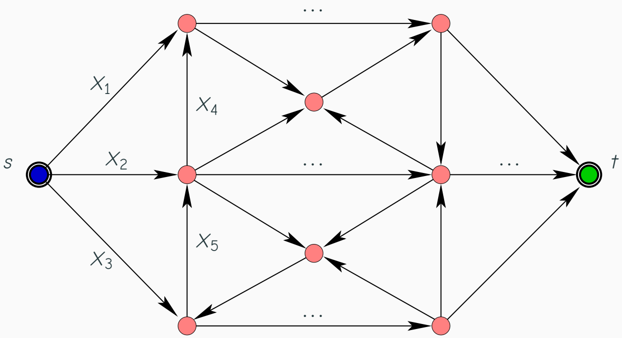
$$\widehat{\zeta} = \frac{R_1}{N_0}\,\frac{R_2}{N_1}\,\frac{R_3}{N_2}\cdots\frac{R_k}{N_{k-1}},$$

where k is the # of thresholds, N_i is the # of clones built after reaching level ℓ_i, and R_j is the # of clones having reached level ℓ_j.

- We want to have (to observe) $R_k > 0$ (enough times). Otherwise, the procedure will return 0, or a too small fraction, typical rare event issue.
- Choosing how many levels to use (parameter k), positioning the levels, selecting the control parameters $N_0, \ldots, N_{k-1}, R_1, \ldots, R_k$ is complicated. There is a large # of possible configurations.
- Computing the variance of the Splitting estimator is too hard. So, there is no theorem allowing to have a theoretical comparison of the different possible configurations.
- There is also no formal result leading to optimal configurations of the procedure (sufficient conditions), except in simple particular cases.
- What people have is more or less simple models where it is possible to find optimal values. Then, they are used as heuristic guidelines to choose all those hyper-parameters in the variant we are running.

- P. Glasserman et al. “Splitting for Rare Event Simulation: Analysis of Simple Cases”. In: Proceedings of the 1996 Winter Simulation Conference. San Diego, California: IEEE Computer Society Press, 1996, pp. 302–308.
- M. J. J. Garvels. “The Splitting Method in Rare Event Simulation”. PhD thesis. Faculty of mathematical Science, University of Twente, The Netherlands, 2000.
- M. J. J. Garvels, D. P. Kroese, and J.–K. C. W. Van Ommeren. “On the Importance Function in Splitting Simulation”. In: European Transactions on Telecommunications 13.4 (2002), pp. 363–371.
- M. Villén-Altamirano and J. Villén-Altamirano. “On the efficiency of RESTART for multidimensional state systems”. In: ACM Transactions on Modeling and Computer Simulation 16 (3 July 2006), pp. 251–279.
- G. Rubino and B. Tuffin (editors and co-authors). Rare Event Simulation Using Monte Carlo Methods. Wiley, 2009.

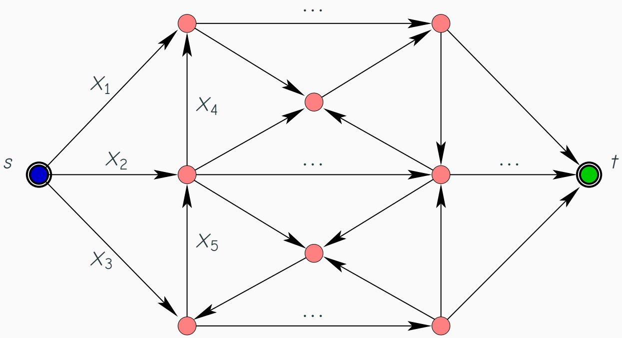
2—Flows



$\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathbf{X})$

- $\mathscr{V}$: set of n nodes, with a source (**blue**) and a terminal (**green**);
- $\mathscr{E}$: set of m arcs;
- $\mathbf{X}$: vector of arc capacities, $(X_1, \ldots, X_m) \geq (0, \ldots, 0)$.

- Flow: a function on the arcs, satisfying local balance at the interior (**red**) nodes.
- Result: the sum of the flows on the arcs leaving s is equal to the sum of the flows of the arcs arriving at t, and it is called the *flow value*.
- $V(X)$: the maximum possible flow value, underlying the dependence on the capacities. Algorithms to compute it: Ford-Fulkerson, Edmonds-Karp, etc.



- The capacity X_i of arc i is now a random variable (links can fail).
- $\mathbf{X} = (X_1, \ldots, X_m)$ is a random vector with values in some $S = (S_1, \ldots, S_m)$.
- We are given a demand d, and the question is to know if the transportation network can transport at least that quantity of fluid.
- The target of the analysis is the estimation of $\zeta = \mathbb{P}\{V(\mathbf{X}) < d\}$.
- If the network is large, ζ will be difficult or impossible to compute (NP hard territory). If the network is highly reliable ($\zeta \ll 1$), Standard (Crude) Monte Carlo will suffer if applied naively.

Our goals:

- To focus on the case of $\zeta \ll 1$ (the rare event case).
- To convert this static model into a dynamic one (a stochastic process) and apply Splitting to it, for the rareness issue.
- To find a Splitting procedure also able to deal with the case of dependent arc failures (this led to use Marshall-Olkin copulas to model the dependencies).

Assume (only for the presentation) that

- the capacities $X_1, \ldots, X_n$ are discrete i.i.d.
- with $\mathbb{P}(X_i = M_j) = p_j$, $j = 1, \ldots, n$ (homogeneous situation),
- $M_1 > M_2 > \ldots > M_n > 0$,
- plus $\mathbb{P}(X_i = 0) = p_0$.

We say that

- if $X_i = 0$, the link is failed,
- if $M_2 \geq X_i \geq M_n$, the link is partially working,
- if $X_i = M_1$, the link is fully operational.

- The homogeneous assumption is just for simplicity in the presentation.
- Everything here also works when each X_i has a different distribution. The notation will need to be $(M_{i,j})$, $(p_{i,j})$, where $i = 1, \ldots, m$ and for component i, $j = 0, 1, 2, \ldots, n_i$, with $1 + n_i = \#$ of possible values of the capacity X_i.

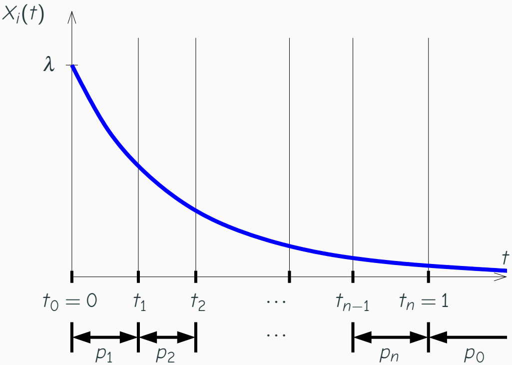
3—From static to dynamic

We build now a dynamics on the same structure.
Basic principles:

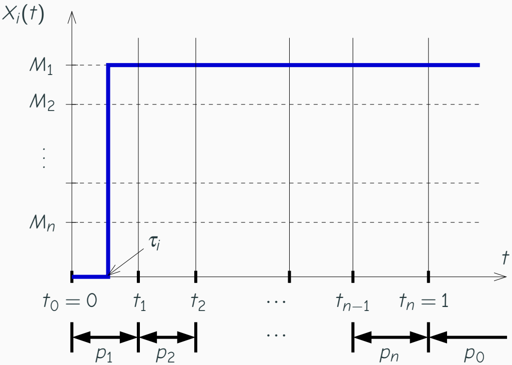
- At time $t = 0$, all links are failed (all the capacities are 0).
- At $t = 0$, we start m artificial "repairing processes" in parallel, for all links, where link i is repaired after an Exponentially distributed random delay τ_i.
- The m r.v.s $\tau_1, \ldots, \tau_m$ are independent.
- Observe that, now, $X = \{X(t),\, t \geq 0\}$.

Our goal:

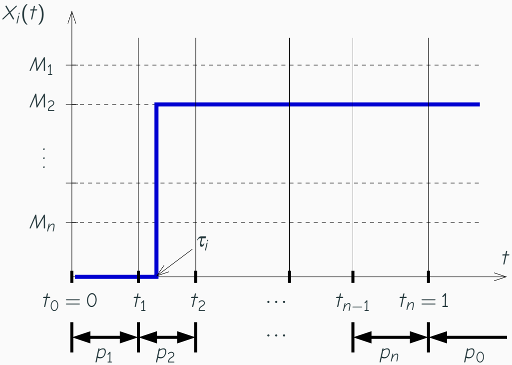
We want that, in the new dynamic system, $\mathbb{P}(V(X(\mathbf{1})) < d) = \zeta$, where ζ is the target defined in the initial static context.



We will choose the rate λ, and the points $t_1, t_2, \ldots, t_{n-1}$ such that $\mathbb{P}(t_{k-1} < \tau_i \leq t_k) = p_k$, $k = 1, \ldots, n$, where $t_0 = 0$ and $t_n = 1$, and $\mathbb{P}(\tau_i > 1) = p_0$.



- If $0 < \tau_i \le t_1$ $X_i(t) : 0 \to M_1$ and keep this value forever,
  If $t_1 < \tau_i \le t_2$ $X_i(t) : 0 \to M_2$ and keep this value forever,
  $\vdots$
  If $t_{n-1} < \tau_i \le 1$ $X_i(t) : 0 \to M_n$ and keep this value forever,
  If $1 < \tau_i$ keep $X_i(t) = 0$ forever.

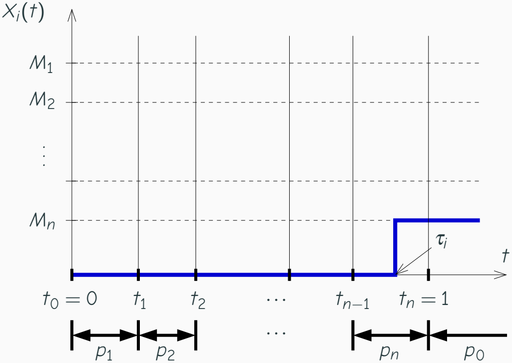


- If $0 < \tau_i \le t_1$ $X_i(t) : 0 \to M_1$ and keep this value forever,
- If $t_1 < \tau_i \le t_2$ $X_i(t) : 0 \to M_2$ and keep this value forever,

$\vdots$

- If $t_{n-1} < \tau_i \le 1$ $X_i(t) : 0 \to M_n$ and keep this value forever,
- If $1 < \tau_i$ keep $X_i(t) = 0$ forever.

3–From static to dynamic
A construction of the Multilevel Creation Process



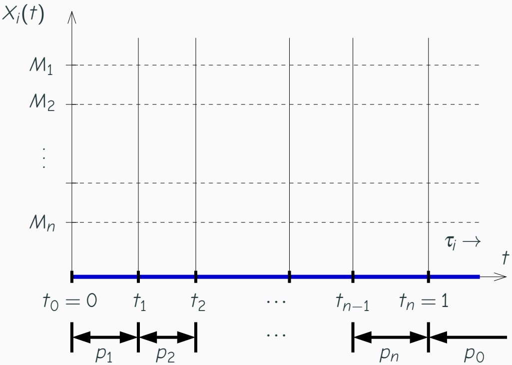
If $0 < \tau_i \le t_1$ $\quad X_i(t) : 0 \to M_1$ and keep this value forever,

If $t_1 < \tau_i \le t_2$ $\quad X_i(t) : 0 \to M_2$ and keep this value forever,

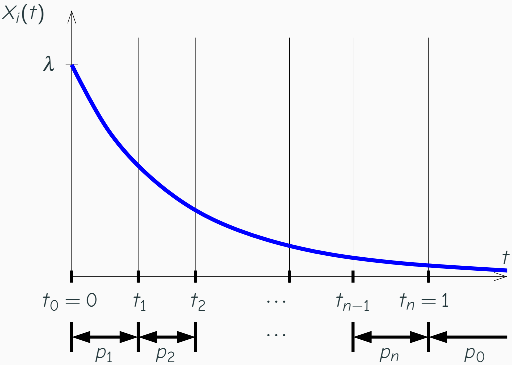
$\vdots$

- If $t_{n-1} < \tau_i \le 1$ $\quad X_i(t) : 0 \to M_n$ and keep this value forever,

If $1 < \tau_i$ $\quad$ keep $X_i(t) = 0$ forever.



If	$0 < \tau_i \le t_1$	$X_i(t) : 0 \to M_1$ and keep this value forever,
If	$t_1 < \tau_i \le t_2$	$X_i(t) : 0 \to M_2$ and keep this value forever,
	$\vdots$	
If	$t_{n-1} < \tau_i \le 1$	$X_i(t) : 0 \to M_n$ and keep this value forever,
▶ If	$1 < \tau_i$	keep $X_i(t) = 0$ forever.

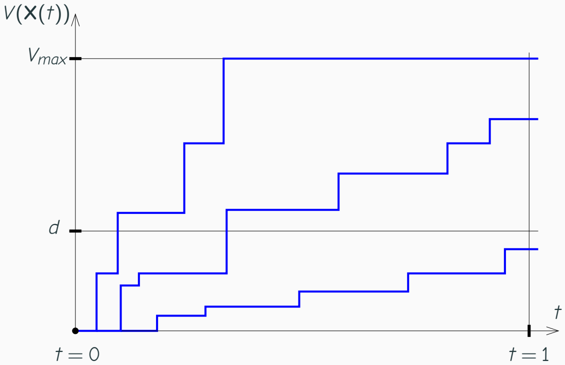


After some algebra, we can take

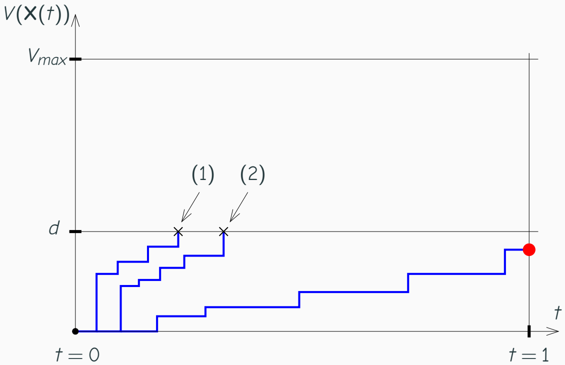
$$\lambda = -\ln(p_0)$$

$$t_k = \frac{\ln(1 - p_1 - p_2 - \ldots - p_k)}{\ln(p_0)}, \qquad k = 1, \ldots, n-1.$$

- When building trajectories, we need to take the τ_is from smallest to largest.
- In other words, we must work with the order statistic of the sequence of repair times, $\tau_{(1)}, \ldots, \tau_{(m)}$.
- For this task, it's more efficient to use the properties of the Exponential, in particular of the min of independent Exponential r.v.s.
- We know that
  - $\tau_{(1)}$ is Exponential with rate $\Lambda = \lambda_1 + \cdots + \lambda_m$,
  - $\operatorname{argmin}\{\tau_i\} = \lambda_i/\Lambda$,
  - knowing that $\tau_{(1)} = \tau_h$, $\tau_{(2)}$ is Exponential with rate $\Lambda - \lambda_h$,
  - etc.

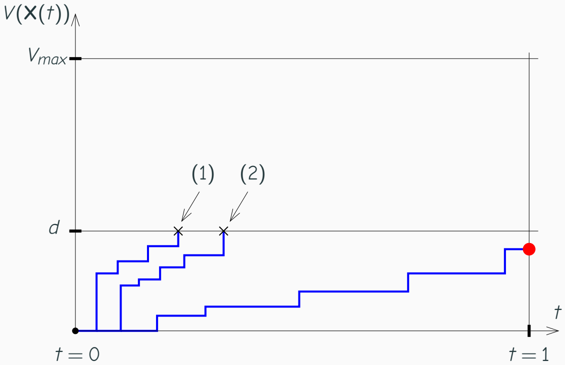


- Following the "repairs", $V(\mathbf{X})$ grows by jumps.
- V_{max} is the max possible value (for instance, the value corresponding to putting all the links at their max capacities).
- The point is to see if $V(\mathbf{X}(1))$ is above or below d.



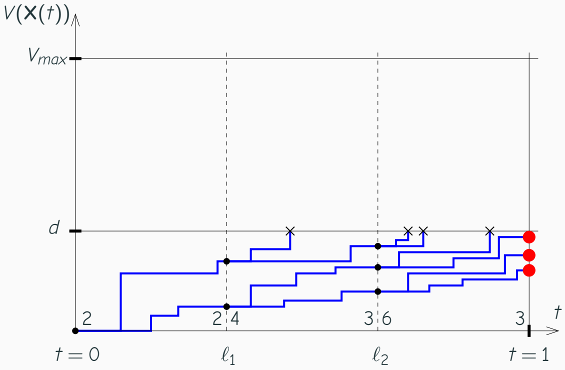
- The trajectories crossing level d before $t = 1$ will be above d at $t = 1$ (this is the typical case).
- After points such as (1) and (2), it is then useless to continue the paths.
- Crude Monte Carlo estimation of ζ is the ratio between the # of red points and the total # of simulated trajectories.

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



- In the highly reliable case, the trajectories will strongly tend to go up (repairs are fast).
- Only very few of them will cross the $t = 1$ border below d (building red points).
- A way of making more efficient the process would be to "stimulate" in some way the trajectories to go to the right (for instance, slowing down the repairs).

4–Splitting
on the artificial dynamic process



- An appropriate application of Splitting requires, once configured,
  - to define the thresholds on the time axis,
  - then to clone the paths that reach next level before crossing the horizontal line "$y = d$".
- The resulting estimation (in the picture) is $\widehat{\zeta} = \frac{2}{2}\,\frac{3}{4}\,\frac{3}{6} = 0.375$

Recall that our ultimate goal is to be able to deal with dependent components.

- We must choose how to model dependencies. One of the easiest ways is to use Marshall–Olkin copulas, based on the idea that failures happen in *shocks*.
- A shock is the simultaneous failure of a subset of components; it's just a way to use data reporting, for instance, correlations between individual failure events.
- The process of going from correlation data to shocks and their associated failure rates is out of the scope of the talk (see the references).
- Given these remarks and after observing that the Creation Process isn't well adapted to this setting, we made the same kind of multi-level extension to the dual Destruction Process (DP).
- In a DP, the m components start operational, and they fail one after the other according to a similar dynamic process as before.
- But moving Splitting to the DP context is not straightforward and we had to make some important changes.

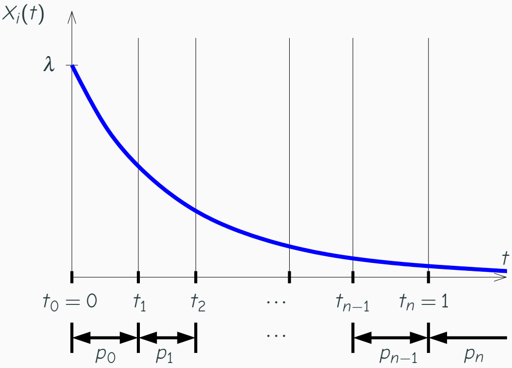
Now, $X_i \in S_i = \{M_n, \ldots, M_2, M_1, 0\}, \quad i = i, \ldots, m$, with

$$X_i = \begin{cases} M_n & \text{w.p. } p_n, \\ & \vdots \\ M_2 & \text{w.p. } p_2, \\ M_1 & \text{w.p. } p_1, \\ 0 & \text{w.p. } p_0, \end{cases} \qquad M_n > \cdots > M_2 > M_1 > 0.$$

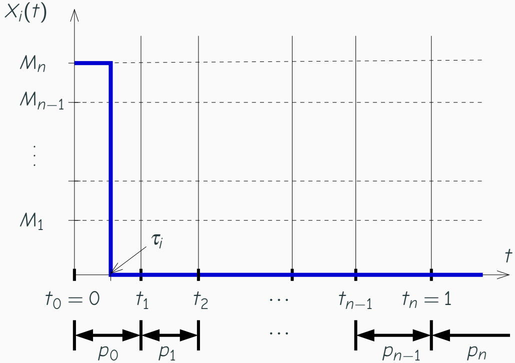
- If $X_i = M_n$, arc i is fully operational.
- If $0 < X_i < M_n$, arc i is partially failed.
- If $X_i = 0$, arc i is totally failed.

All links start fully up (value M_n in the homogeneous case), and link i will fail at time τ_i, assumed to be Exponentially distributed.

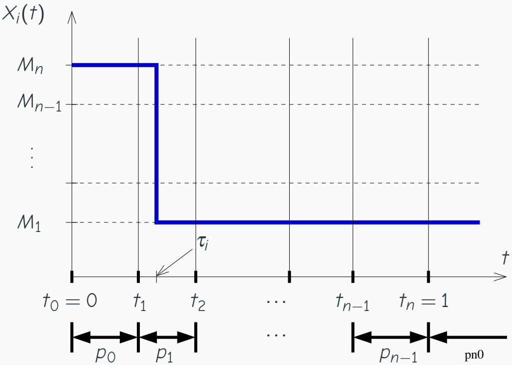
We call *failure* the move from M_n to some M_k, $k \leq n-1$, or to 0.



We will choose the rate λ, and the points $t_1, t_2, \ldots, t_{n-1}$ such that $\mathbb{P}(t_k < \tau_i \leq t_{k+1}) = p_k$, $k = 0, \ldots, n-1$, where $t_0 = 0$ and $t_n = 1$, and $\mathbb{P}(\tau_i > 1) = p_n$.



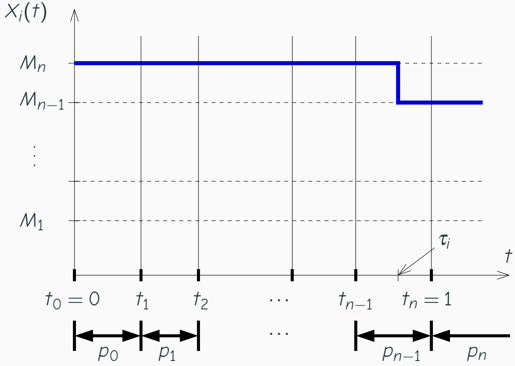
- If $0 < \tau_i \le t_1$ $\quad X_i(t) : M_n \to 0$ and keep this value forever,
  If $t_1 < \tau_i \le t_2$ $\quad X_i(t) : M_n \to M_1$ and keep this value forever,
  $\vdots$
  If $t_{n-1} < \tau_i \le 1$ $\quad X_i(t) : M_n \to M_{n-1}$ and keep this value forever,
  If $1 < \tau_i$ $\quad$ keep $X_i(t) = M_n$ forever.



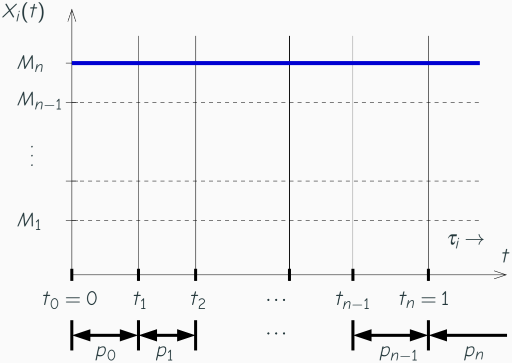
- If $0 < \tau_i \le t_1$ $X_i(t) : M_n \to 0$ and keep this value forever,
- If $t_1 < \tau_i \le t_2$ $X_i(t) : M_n \to M_1$ and keep this value forever,

  $\vdots$

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



If	$0 < \tau_i \le t_1$	$X_i(t) : M_n \to 0$ and keep this value forever,
If	$t_1 < \tau_i \le t_2$	$X_i(t) : M_n \to M_1$ and keep this value forever,
	$\vdots$	
▶ If	$t_{n-1} < \tau_i \le 1$	$X_i(t) : M_n \to M_{n-1}$ and keep this value forever,
If	$1 < \tau_i$	keep $X_i(t) = M_n$ forever.

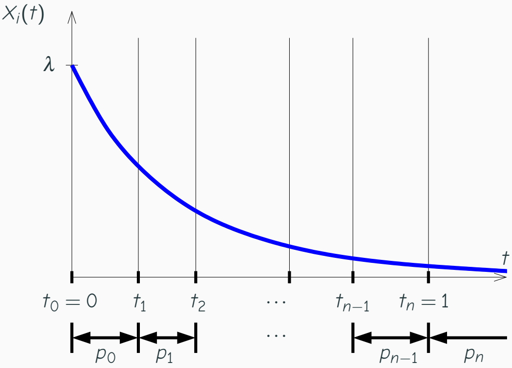


If $0 < \tau_i \le t_1$ $X_i(t) : M_n \rightarrow 0$ and keep this value forever,
If $t_1 < \tau_i \le t_2$ $X_i(t) : M_n \rightarrow M_1$ and keep this value forever,

$\vdots$

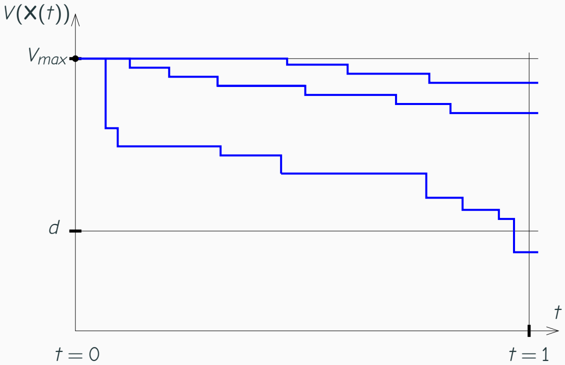
If $t_{n-1} < \tau_i \le 1$ $X_i(t) : M_n \rightarrow M_{n-1}$ and keep this value forever,

- If $1 < \tau_i$ keep $X_i(t) = M_n$ forever.

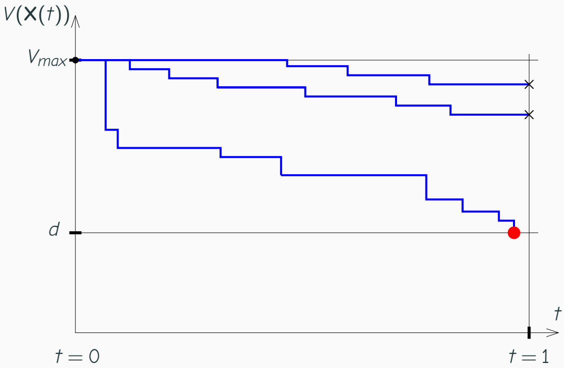


Values:

$$\lambda = -\ln(p_n),$$
$$t_k = \frac{\ln(1 - p_0 - p_1 - \ldots - p_{k-1})}{\ln(p_n)}, \quad k = 1, \ldots, n-1.$$

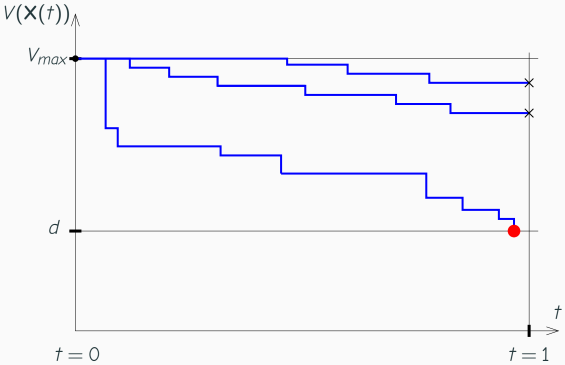


- Following the "failures", $V(\mathbf{X})$ decreases by jumps.
- V_{max} is the max possible value (for instance, the value corresponding to putting all the links at their max capacities).
- The point is to see if $V(\mathbf{X}(1))$ is above or below d.

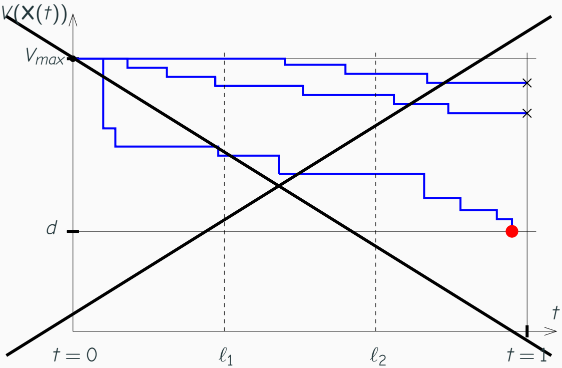


- The trajectories crossing level d before $t = 1$ will be below d at $t = 1$ (this is the typical case).
- After red points it is then useless to continue the paths.
- Crude Monte Carlo estimation of ζ is the ratio between the # of red points and the total # of simulated trajectories. Here,

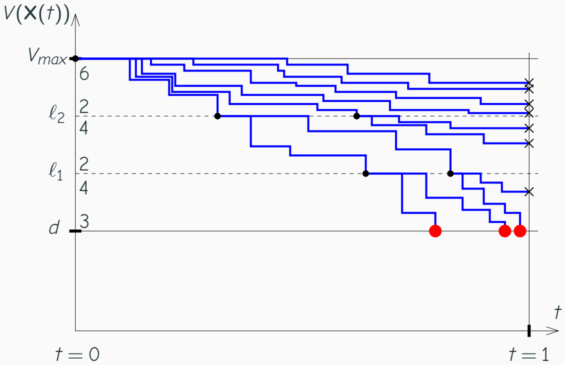
$$\widehat{\zeta} = \frac{1}{3} = 0.333$$



- In the highly reliable case, the trajectories will rarely go down.
- Only very few of them will cross the "$y = d$" line before $t = 1$ (building red points).
- A way of making more efficient the process would be to "stimulate" in some way the trajectories to go down.

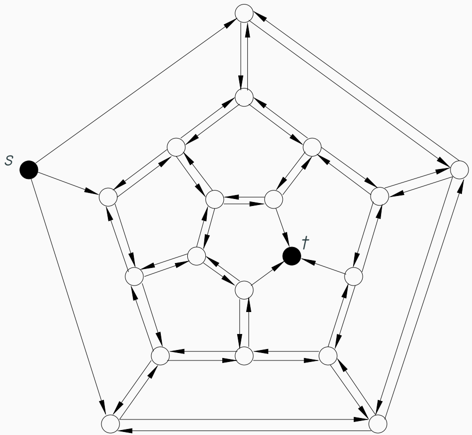


- All this means that a similar use of the Splitting idea shown before doesn't work.
- Using thresholds on the time axis will push the clones to the right, and we want them to go down.



- The idea is to put the thresholds on the vertical axis, on the $V(\mathbf{X}(t))$ values,
- and to clone the trajectories from the crossing points when they happen before $t = 1$.
- In the picture, we then have $\widehat{\zeta} = \dfrac{2}{6}\,\dfrac{2}{4}\,\dfrac{3}{4} = 0.125$

- In the talk, I omitted details about the implementation of the Splitting algorithms.
- In particular, the hyper-parameters issues are not described, and we don't discuss the different versions of Splitting that can be used. In the paper, we followed some guidelines of this type, and used pilot runs for fine tuning them, before simulating.
- Another important issue in this area is variance control, because we lack results concerning the variance of the different versions of the method. As said before, we only have optimality results on pretty simple models providing guidelines on the design of the procedures.
- Another important practical issue is data leading to the shock-based model for handling dependencies. There is a reference at the end where we explored this issue with some mathematical details about the technical side.



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

$\widehat{\zeta}$	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

$\widehat{\zeta}$: the target (an unavailability metric).

RE: Relative Error, $\mathrm{SD}(\widehat{\zeta})/\mathbb{E}(\widehat{\zeta})$

Th: number of thresholds.

N: total number of paths.

t: simulation time in sec.

d: demand value.

Last comments

- From our viewpoint, the most interesting aspect of this proposal is the fact that it fits with the problem of relaxing the usual independent components assumption in the models.
- The other point to underline is the good behavior of the method when the failure of the system becomes very rare.
- It will be useful to try Importance Sampling on this problem, perhaps the so-called Zero Variance sub-family, for comparison purposes.

Some references

- M. Lomonosov. "On Monte Carlo estimates in network reliability". In: Prob. in Eng. and Informational Sciences 8 (1994), pp. 245–264.

  The Creation Process is defined in this paper.

- Héctor Cancela, Leslie Murray, and Gerardo Rubino. "Efficient Estimation of Stochastic Flow Network Reliability". In: IEEE Transactions on Reliability 68, 3 (Sep. 2019), pp. 954—970.

  This paper proposes the Multilevel extension of the initial Creation Process and the associated Splitting approach to the flow problem described here.

- O. Matus, E. Moreno, J. Barrera and G. Rubino. "On the Marshall–Olkin Copula Model for Network Reliability Under Dependent Failures". In: IEEE Transactions on Reliability 68, 2 (June 2019), pp. 451–461.

  This paper illustrates the use of the Marshall–Olkin copulas in the area and the benefits in accuracy this tool offers.

- Héctor Cancela, Leslie Murray, and Gerardo Rubino. "**Reliability Estimation for Stochastic Flow Networks with Dependent Arcs**". In: IEEE Transactions on Reliability (2022), to appear (the revised version was accepted this week).

  This paper corresponds to the talk. It has some numerical results on a couple of models.

- Zdravko I. Botev, Pierre L'Ecuyer, Gerardo Rubino, Richard Simard, Bruno Tuffin. "Static Network Reliability Estimation via Generalized Splitting". In: INFORMS J. of Computing (2013), 25, 1.

  A related work on applying Splitting to a more classical network reliability problem.