



**HAL**  
open science

# The Multi-Level Creation Process in Flow Network Reliability Estimation

Héctor Cancela, Leslie Murray, Gerardo Rubino

► **To cite this version:**

Héctor Cancela, Leslie Murray, Gerardo Rubino. The Multi-Level Creation Process in Flow Network Reliability Estimation. MCQMC'18 - 13th International Conference in Monte Carlo & Quasi-Monte Carlo Methods in Scientific Computing, Jul 2018, Rennes, France. pp.1-42. hal-01962927

**HAL Id: hal-01962927**

**<https://inria.hal.science/hal-01962927v1>**

Submitted on 20 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THE MULTI-LEVEL CREATION PROCESS IN FLOW NETWORK RELIABILITY ESTIMATION

Héctor Cancela <sup>1</sup>   Leslie Murray <sup>2</sup>   Gerardo Rubino (**Speaker**) <sup>3</sup>

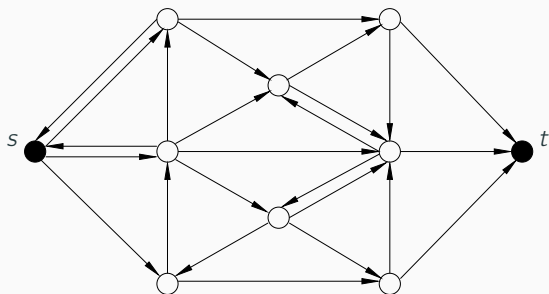
<sup>1</sup>Facultad de Ingeniería, UDELAR, Montevideo, Uruguay

<sup>2</sup>Facultad de Ingeniería, UNR, Rosario, Argentina

<sup>3</sup>INRIA Rennes, France

July 6, 2018

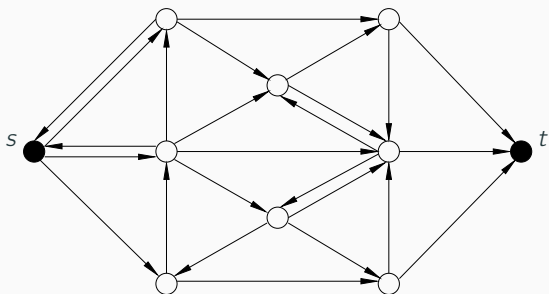
# INTRODUCTION: FLOW NETWORK



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

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

# INTRODUCTION: STOCHASTIC FLOW NETWORK



- The capacities  $X_i$ ,  $i = 1, \dots, m$ , change due to failures
- $\mathbf{X} = (X_1, \dots, X_m)$ , is a random vector in  $\Omega = (\Omega_1, \dots, \Omega_m)$
- $\zeta = \mathbb{P}\{M(\mathbf{X}) < T\}$ . If the network is highly reliable,  $\zeta \ll 1$

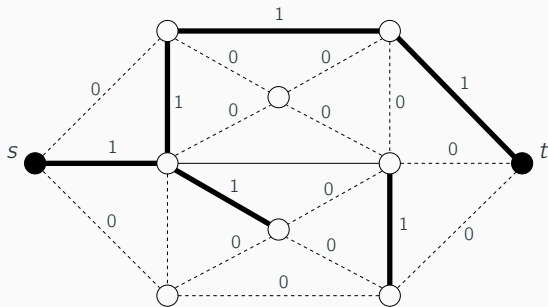
## AIM OF THIS TALK

To introduce an efficient method for the estimation of  $\zeta$ , when  $\zeta \ll 1$

- ① NETWORK STATIC MODEL (REVIEW)
  - Basic Setting / Crude Monte Carlo
  - The Creation Process
  - Splitting on the the Creation Process
- ② STOCHASTIC FLOW NETWORK MODEL
  - Multi-Level Creation Process
  - Splitting on the Multi-Level Creation Process
- ③ EXPERIMENTAL RESULTS
  - Efficiency Analysis
- ④ CONCLUSIONS AND LINES OF FUTURE WORK

# NETWORK STATIC MODEL

$$\mathbf{X} = (X_1, X_2, \dots, X_m) \rightarrow X_i = \begin{cases} 1 \text{ w.p. } r_i & i^{\text{th}} \text{ link operative} \\ 0 \text{ w.p. } q_i = 1 - r_i & i^{\text{th}} \text{ link failed} \end{cases}$$



$$\phi(\mathbf{X}) = \begin{cases} 1 & \text{if } s \text{ and } t \text{ are not connected} \\ 0 & \text{if } s \text{ and } t \text{ are connected} \end{cases} \rightarrow \zeta = \mathbb{P}\{\phi(\mathbf{X}) = 0\}$$

# CRUDE MONTE CARLO

- 1 Sample  $N$  independent instances of  $\mathbf{X}$  :  $\mathbf{X}^{(i)}$ ,  $i = 1, \dots, N$
- 2 Compute  $\phi(\mathbf{X}^{(i)})$ ,  $i = 1, \dots, N$
- 3 Estimate  $\zeta$  as:

$$\hat{\zeta} = \frac{1}{N} \sum_{i=1}^N \phi(\mathbf{X}^{(i)})$$

$$\text{RE} \triangleq \frac{\sqrt{\text{V}\{\hat{\zeta}\}}}{\mathbb{E}\{\hat{\zeta}\}} = \frac{\sqrt{\zeta(1-\zeta)}}{\sqrt{N}\zeta} \approx \frac{1}{\sqrt{N}\zeta} \rightarrow \infty \quad \text{if } \zeta \rightarrow 0 \text{ with } N \text{ fixed}$$

$\uparrow$   
if  $\zeta \ll 1$

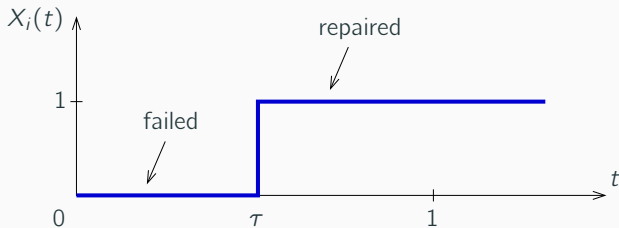
CRUDE MC DOES NOT SOLVE THE PROBLEM...

Rare event problem.

# THE CREATION PROCESS

Time enters:  $\mathbf{X} \rightarrow \mathbf{X}(t) = (X_1(t), X_2(t), \dots, X_m(t))$

For each component, and for the whole system,

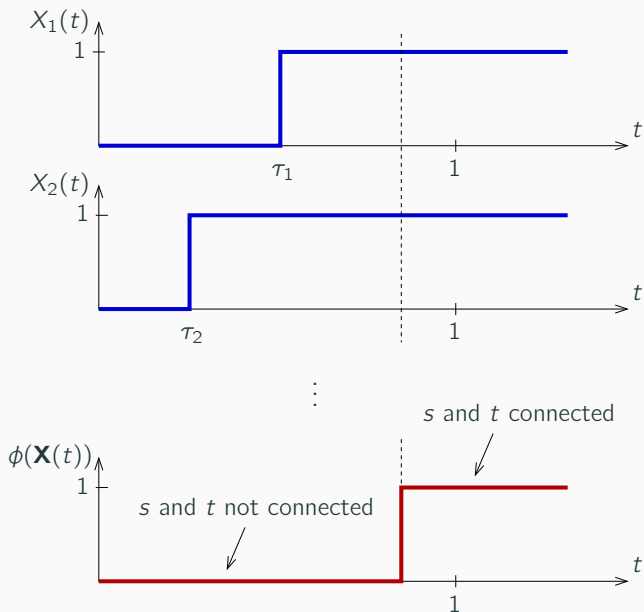


## BASIC IDEA

If links are repaired in a time proportional to their own single reliabilities,  $s$  and  $t$  become connected in a time proportional to the network's reliability



# THE CREATION PROCESS



# THE CREATION PROCESS

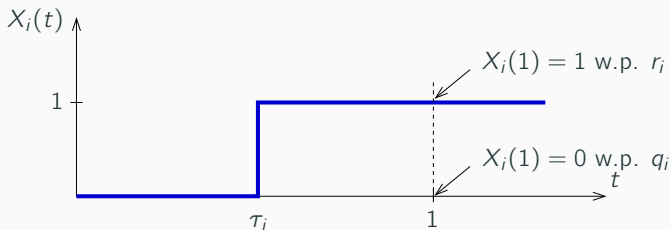
Let  $\tau_i$  be the repair time of component  $i$ , with

- $\tau_i \sim \exp(\lambda_i)$ ,  $i = 1, \dots, m$
- $\lambda_i = -\ln(q_i)$

Then:

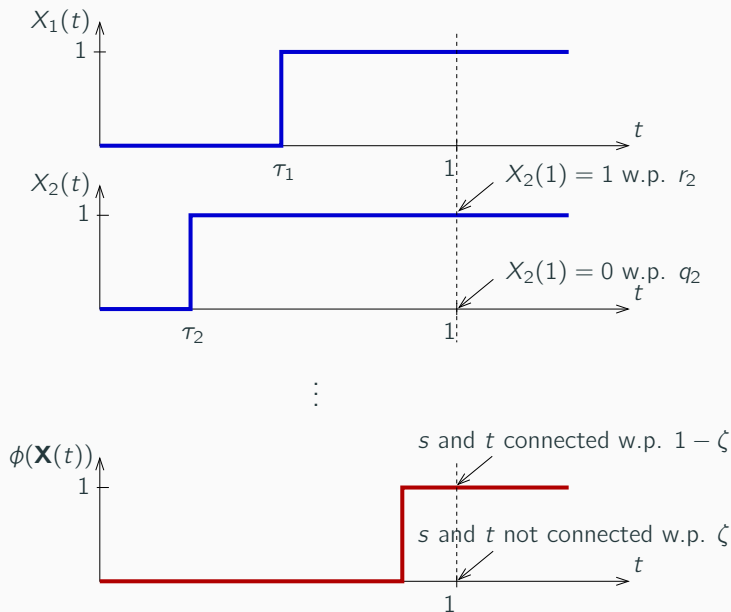
$$\mathbb{P}\{X_i(1) = 1\} = \mathbb{P}\{\tau_i \leq 1\} = 1 - e^{-\lambda_i} = 1 - e^{\ln(q_i)} = r_i$$

$$\mathbb{P}\{X_i(1) = 0\} = \mathbb{P}\{\tau_i > 1\} = e^{-\lambda_i} = e^{\ln(q_i)} = q_i$$



Finally, if  $\mathbb{P}\{X_i(1) = 0\} = q_i$ ,  $i = 1, \dots, m$ , then  $\mathbb{P}\{\phi(\mathbf{X}(1) = 0)\} = \zeta$

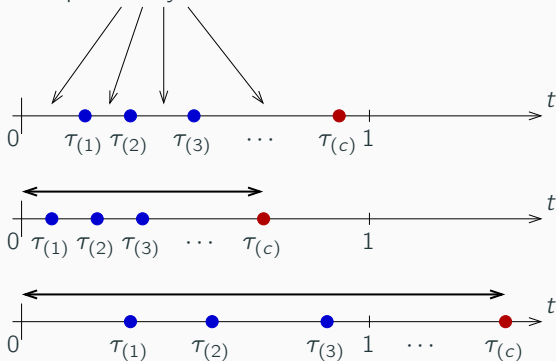
# THE CREATION PROCESS



# THE CREATION PROCESS

Let  $\tau_c$  be the repair time of the system. and consider the following point process.

times between repairs are also exponentially distributed



## ALGORITHM

Sample sequences  $\{\tau_{(1)}, \tau_{(2)}, \dots, \tau_{(c)}\}$ , see them as trajectories in a one-dimensional space and reduce the problem to:  $\zeta = \mathbb{P}\{\tau_c > 1\}$

# THE CREATION PROCESS / CRUDE MC



- 1 Sample  $N$  times the sequence  $\{\tau_{(1)}, \tau_{(2)} \cdots, \tau_{(c)}\}$
- 2 From each one of them, read the value of the random variable  $\tau_{(c)}$  and compute the function:

$$I = \begin{cases} 1 & \text{if } \tau_{(c)} > 1, \\ 0 & \text{otherwise} \end{cases} \quad \rightarrow \quad I^{(i)}, \quad i = 1, \dots, N$$

- 3 Estimate  $\zeta$  as:

$$\hat{\zeta} = \frac{1}{N} \sum_{i=1}^N I^{(i)}$$

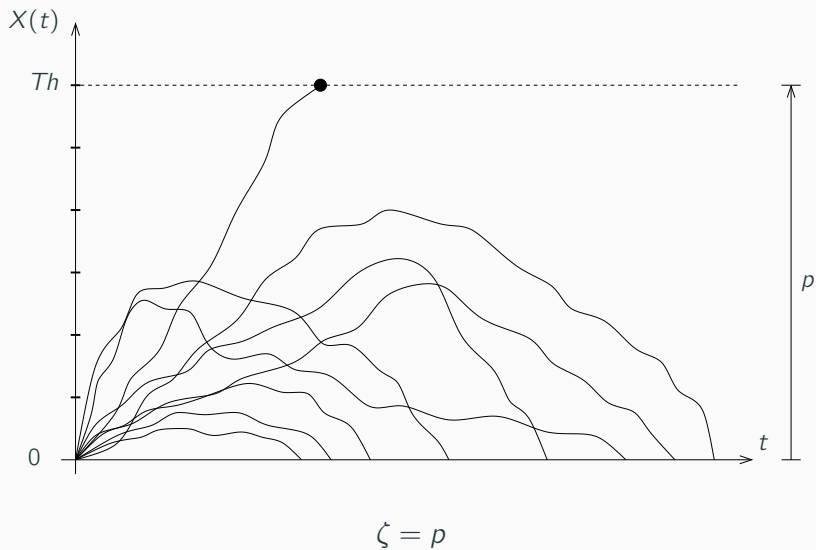
STILL THE SAME CRUDE MONTE CARLO ESTIMATION...

It has to be improved by means of a variance reduction method

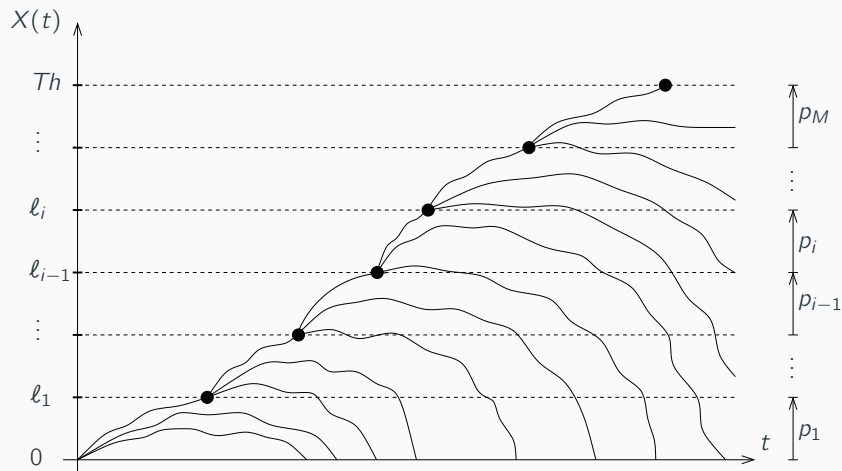
# METHODS BASED ON THE CREATION PROCESS



# SPLITTING



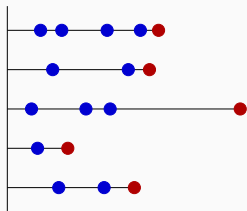
# SPLITTING



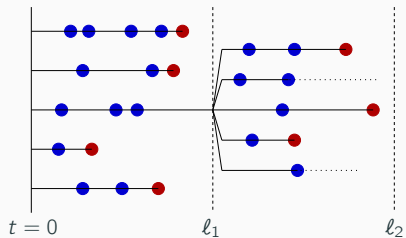
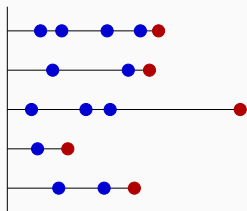
$$\zeta = \prod_{i=1}^M p_i \quad \rightarrow \quad \hat{\zeta} = \prod_{i=1}^M \hat{p}_i$$



# SPLITTING ON THE CREATION PROCESS



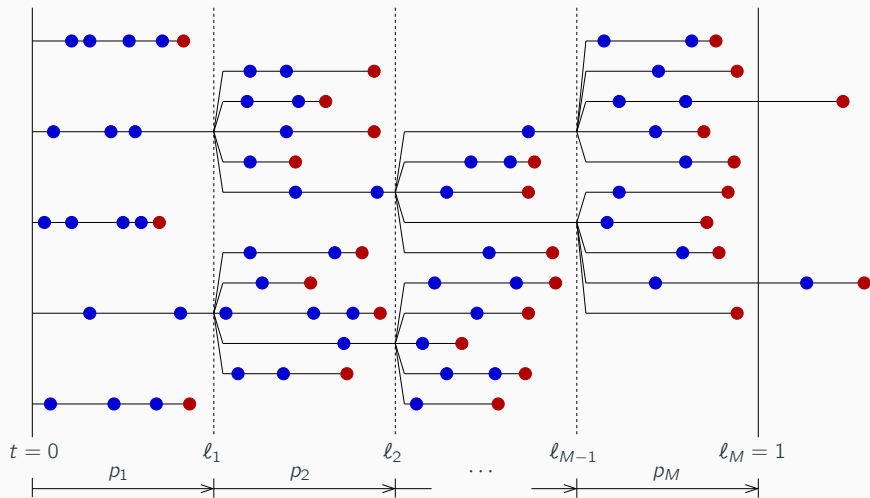
# SPLITTING ON THE CREATION PROCESS



...

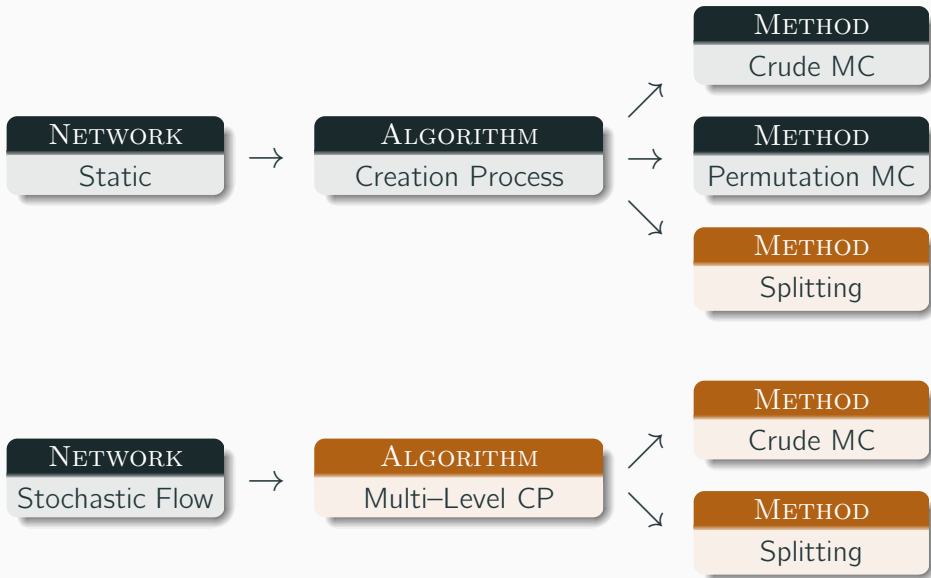
$l_M = 1$

# SPLITTING ON THE CREATION PROCESS



$$\zeta = \prod_{i=1}^M p_i \quad \rightarrow \quad \hat{\zeta} = \prod_{i=1}^M \hat{p}_i$$

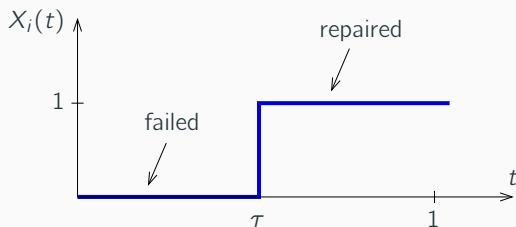
# CREATION PROCESS / MULTI-LEVEL CP



# MULTI-LEVEL CP

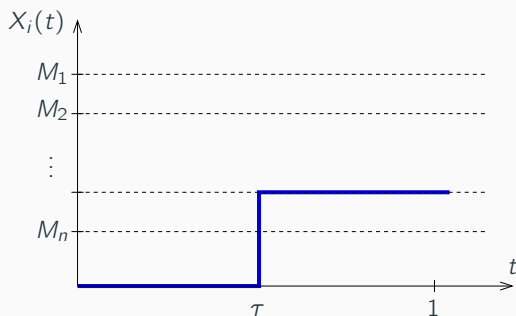
Static Network Model:

$$X_i = \begin{cases} 1 & \text{w.p. } r_i, \\ 0 & \text{w.p. } q_i \end{cases}$$

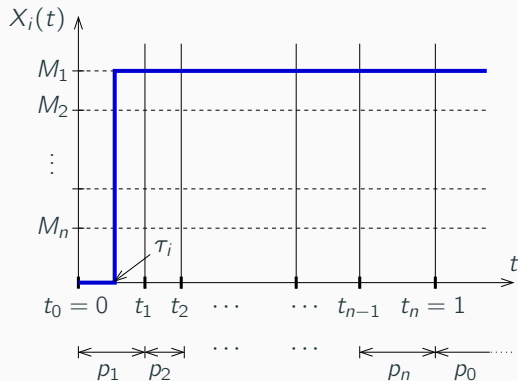


Stochastic Flow Network:

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

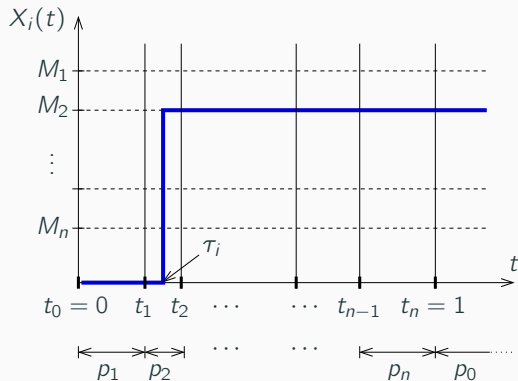


# MULTI-LEVEL CP



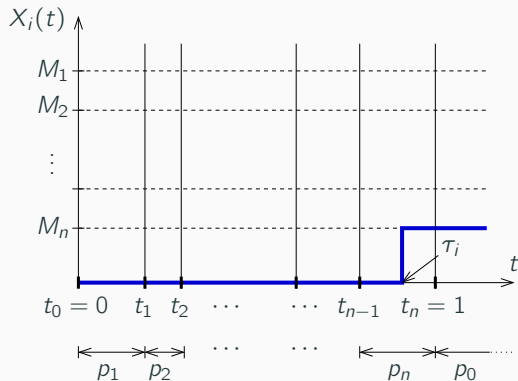
- If  $0 < \tau_i \leq t_1$  set  $X_i(t) = M_1$  and keep this value forever,
- If  $t_1 < \tau_i \leq t_2$  set  $X_i(t) = M_2$  and keep this value forever,
- $\vdots$
- If  $t_{n-1} < \tau_i \leq t_n$  set  $X_i(t) = M_n$  and keep this value forever,
- If  $1 < \tau_i$  leave  $X_i(t) = 0$  forever.

# MULTI-LEVEL CP



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

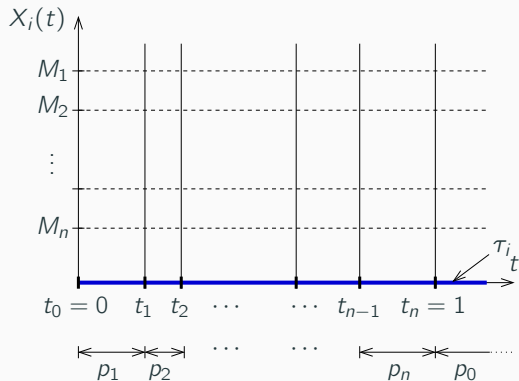
# MULTI-LEVEL CP



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



# MULTI-LEVEL CP



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

As  $\tau$  is exponentially distributed:

$$f_{\tau}(t) = \lambda e^{-\lambda t} \text{ and } F_{\tau}(t) = 1 - e^{-\lambda t}.$$

Two important issues related to  $\tau$  are not yet defined:

- 1 Which is the rate,  $\lambda$ , of its distribution?

$$\mathbb{P}\{\tau > 1\} = 1 - F_{\tau}(1) = 1 - (1 - e^{-\lambda}) = e^{-\lambda} = p_0$$

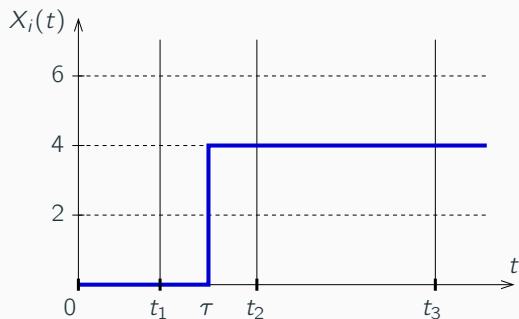
$$\text{Then: } \lambda = -\ln(p_0)$$

- 2 Which are the values of the times  $t_1, t_2, \dots, t_{n-1}$ ?

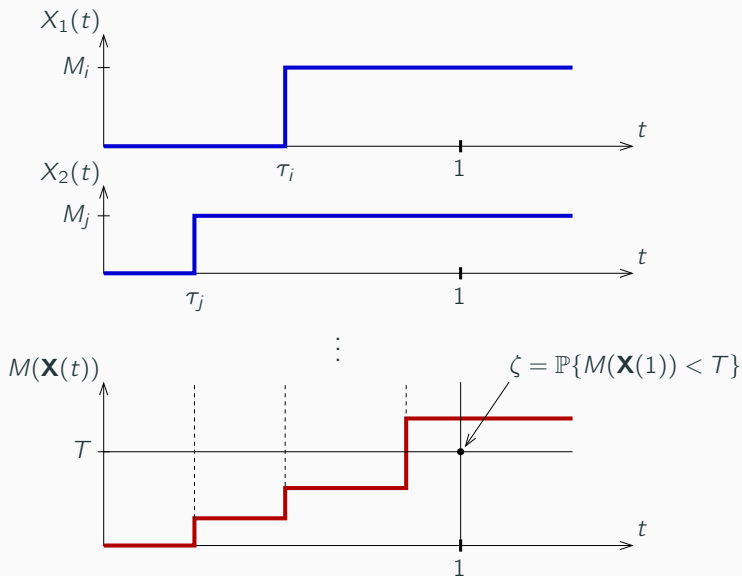
$$\mathbb{P}\{t_{i-1} < \tau \leq t_i\} = p_i \rightarrow t_i = \frac{\ln(1 - p_1 - \dots - p_i)}{\ln(p_0)}$$

# MULTI-LEVEL CP

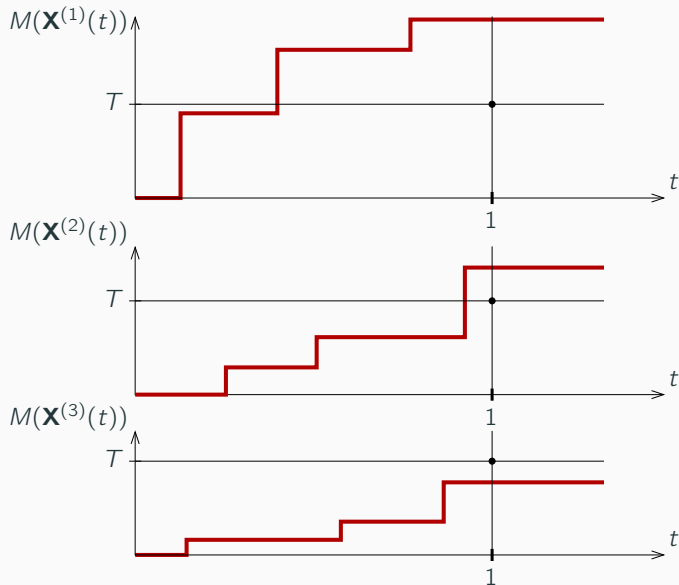
$$X_i = \begin{cases} 0 & \text{w.p. } 0.05 \\ 6 & \text{w.p. } 0.48 \\ 4 & \text{w.p. } 0.29 \\ 2 & \text{w.p. } 0.18 \end{cases} \rightarrow \begin{cases} \lambda = 2.996 \\ t_1 = 0.218 \\ t_2 = 0.491 \\ t_3 = 1.000 \end{cases}$$



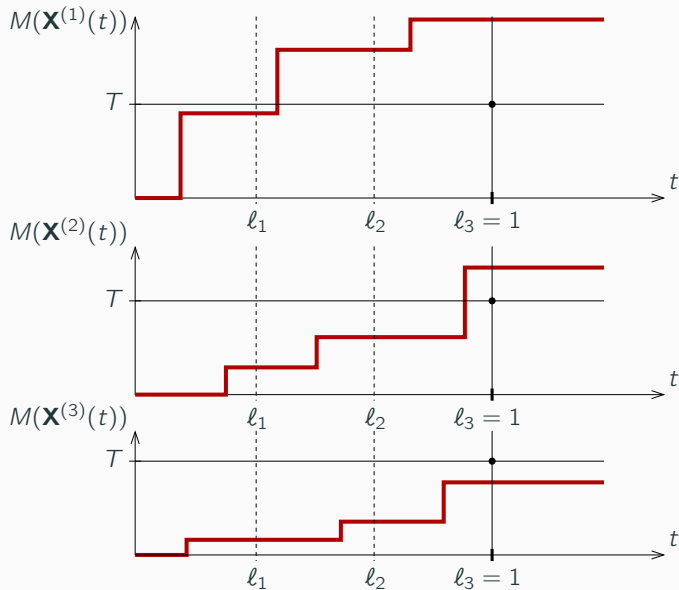
# SPLITTING ON THE MULTI-LEVEL CP



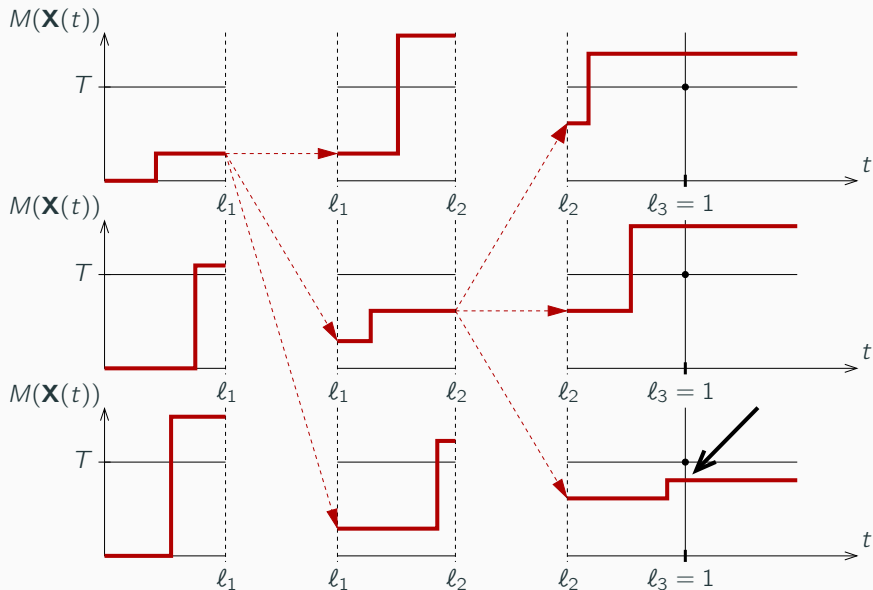
# SPLITTING ON THE MULTI-LEVEL CP



# SPLITTING ON THE MULTI-LEVEL CP



# SPLITTING ON THE MULTI-LEVEL CP



So far, links' capacities were modelled in terms of a discrete distribution,

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

but the model can be changed by another one in which the non-zero capacities follow a continuous distribution like, for example:

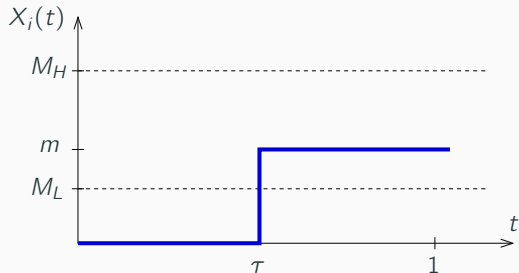
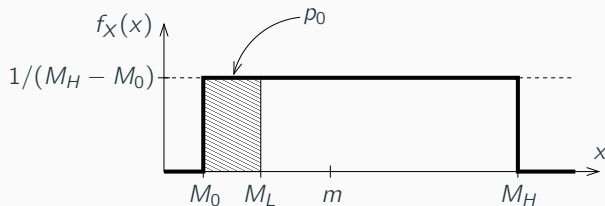
- $X_i = 0$  with probability  $p_0$
- $X_i$  uniformly distributed between  $M_L$  and  $M_H$ , whenever  $X_i \neq 0$

$$X_i = B_i \times U_i$$

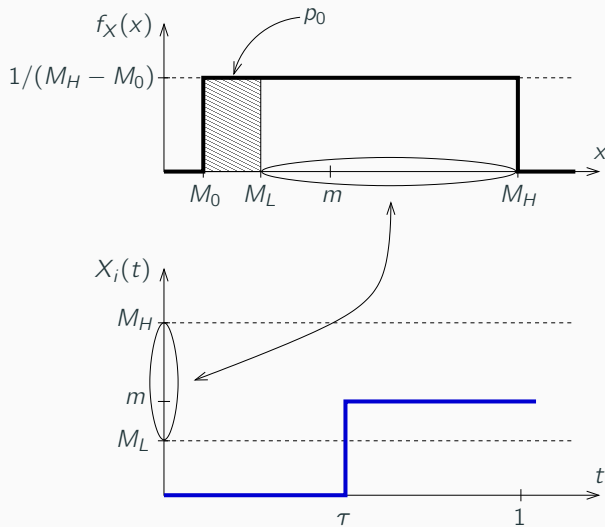
where  $B_i \sim \text{Ber}(1 - p_0)$  and  $U_i \sim \text{Unif}(M_L, M_H)$



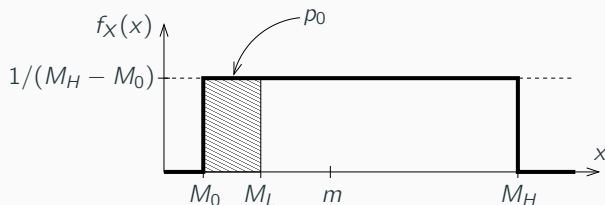
# MULTI-LEVEL CP / CONTINUOUS CAPACITIES



# MULTI-LEVEL CP / CONTINUOUS CAPACITIES



# MULTI-LEVEL CP / CONTINUOUS CAPACITIES



Time  $\tau$  is still exponentially distributed and,

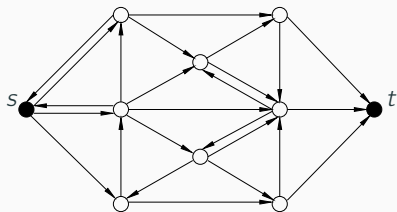
- 1 The rate,  $\lambda$ , of its distribution is:

$$\lambda = -\ln(p_0) = -\ln\left(\frac{M_L - M_0}{M_H - M_0}\right)$$

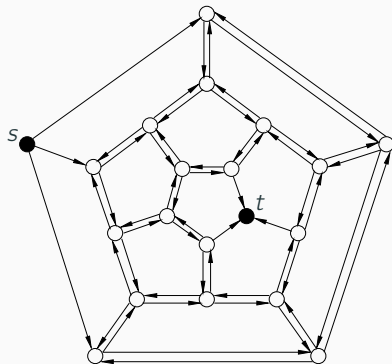
- 2 The capacity,  $m$ , for any sampled time,  $\tau$ , is:

$$m = M_0 + (M_H - M_0)e^{-\lambda\tau}$$

# BENCHMARK NETWORKS



*Fishman Network*



Dodecahedron

# EFFICIENCY ANALYSIS I

Network: *Fishman* Network

Number of Replications:  $10^2$  (effort per level =  $10^3$ )

Links' Capacities:  $X_i = 0$  w.p.  $p_0$  and, while not zero,  $\text{Unif}(100, 200)$

Results:  $\hat{\zeta}$ , RE and *Speedup*

	$T = 300$	$T = 250$	$T = 200$	$T = 150$	$T = 100$
$p_0 = 0.1$	3.18E-01	1.39E-01	6.32E-02	3.54E-02	2.58E-03
$p_0 = 0.01$	2.99E-02	8.18E-03	5.85E-04	3.02E-04	1.92E-06
$p_0 = 0.001$	3.01E-03	7.56E-04	5.88E-06	3.00E-06	1.94E-09
$p_0 = 0.0001$	3.02E-04	7.39E-05	5.89E-08	2.99E-08	1.94E-12
$p_0 = 0.1$	0.42%	0.63%	0.80%	0.80%	1.07%
$p_0 = 0.01$	0.84%	1.30%	1.52%	1.40%	2.25%
$p_0 = 0.001$	1.14%	1.34%	1.73%	1.70%	2.26%
$p_0 = 0.0001$	1.18%	1.74%	1.85%	1.79%	2.57%
$p_0 = 0.1$	0.5	0.5	0.8	1.3	8.6
$p_0 = 0.01$	1.4	2.2	15.4	37.4	2,317.6
$p_0 = 0.001$	5.4	13.3	936.3	1,702.9	1,412,607.4
$p_0 = 0.0001$	42.7	72.4	57,604.2	112,470.0	851,348,214.3

# EFFICIENCY ANALYSIS II

Network: Dodecahedron

Number of Replications:  $10^2$  (effort per level =  $10^3$ )

Links' Capacities:  $X_i = 0$  w.p.  $p_0$  and, while not zero, Unif(100, 200)

Results:  $\hat{\zeta}$ , RE and *Speedup*

	$T = 300$	$T = 250$	$T = 200$	$T = 150$	$T = 100$
$p_0 = 0.1$	3.52E-01	1.60E-01	7.18E-02	4.13E-02	2.93E-03
$p_0 = 0.01$	2.98E-02	8.27E-03	5.96E-04	3.06E-04	2.05E-06
$p_0 = 0.001$	3.00E-03	7.57E-04	6.12E-06	3.01E-06	1.97E-09
$p_0 = 0.0001$	3.02E-04	7.52E-05	5.90E-08	2.98E-08	1.98E-12
$p_0 = 0.1$	0.33%	0.60%	0.75%	0.83%	1.06%
$p_0 = 0.01$	0.94%	1.31%	1.28%	1.43%	2.43%
$p_0 = 0.001$	1.16%	1.53%	1.83%	1.92%	2.21%
$p_0 = 0.0001$	1.27%	1.83%	1.87%	1.89%	2.67%
$p_0 = 0.1$	0.6	0.5	0.8	1.0	9.1
$p_0 = 0.01$	1.1	2.2	22.4	33.1	2,014.2
$p_0 = 0.001$	5.3	11.0	756.6	1,369.4	1,523,596.1
$p_0 = 0.0001$	32.9	66.3	55,301.5	107,402.7	635,779,816.5

# COMPARISON TO OTHER PAPERS' RESULTS

Network: Dodecahedron

Number of Replications:  $6 \times 10^6$  (\*)

Links' Capacities:  $X_i = 0$  w.p.  $q$  and,  $X_i = 1$  w.p.  $1 - q$

Permutation Monte Carlo

$q$	$10^{-6}$	$10^{-5}$	$10^{-4}$	$10^{-3}$	$10^{-2}$	0.10	0.15
$\hat{\zeta}$	2.99E-06	3.02E-05	3.00E-04	3.00E-03	3.06E-02	3.57E-01	5.54E-01
RE	2.08%	1.94%	1.53%	2.11%	1.09%	0.31%	0.28%

Splitting on the Multi-Level CP

$q$	$10^{-6}$	$10^{-5}$	$10^{-4}$	$10^{-3}$	$10^{-2}$	0.10	0.15
$\hat{\zeta}$	3.01E-06	3.00E-05	2.99E-04	3.00E-03	3.07E-02	3.58E-01	5.55E-01
RE	0.76%	0.62%	0.48%	0.38%	0.22%	0.08%	0.05%

(\*) The number of times each method applies the Ford Fulkerson algorithm.

- Multi-Level CP is a generalization of the classical Creation Process.
- Multi-Level CP performs as the base for a Splitting application (Permutation algorithms are not possible).
- Splitting on the Multi-Level CP is a new proposal, reason why, more insightful analysis (variance, error, robustness, . . .) are yet to be done.
- Splitting on the Multi-Level CP performs very efficiently when networks are highly reliable. Experimental results are very encouraging.



# INCREMENTAL FORD FULKERSON

- Every time a threshold  $\ell_i$  is crossed, the condition  $M(\mathbf{X}(\ell_i)) < T$  is checked by means of the Ford Fulkerson algorithm.
- Ford Fulkerson algorithm proceeds iteratively, *flooding* the links up to the maximum possible flow from  $s$  to  $t$ .<sup>(\*)</sup>
- At the crossing of threshold  $\ell_1$  links are *empty* and they are *flooded* up to an amount  $\mathbf{L}_1 = (L_1, \dots, L_m)$  of flow.
- At the crossing of threshold  $\ell_2$  (for all the created trajectories) flows are incremented, starting all of them from  $\mathbf{L}_1$ .

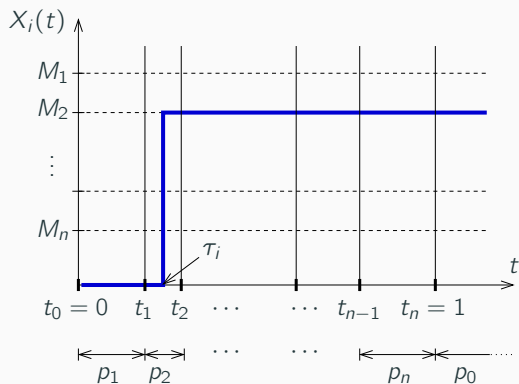
## ALGORITHM IMPROVEMENT

- In standard Ford Fulkerson flows increase, always starting from zero, over empty links.
- In *Incremental Ford Fulkerson*, at every Splitting stage, flows increase starting from the values reached in the previous stage.

*Incremental Ford Fulkerson* significantly reduces the simulation time.

<sup>(\*)</sup>It is an imaginary flow, only for the purposes of the Ford Fulkerson algorithm.

# CAPACITY SAMPLING PLAN

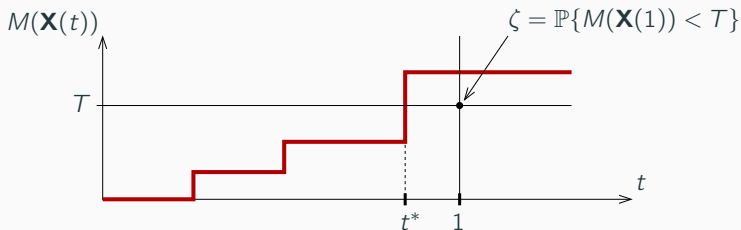


- Largest *capacity*  $M_1$  comes from the lowest times,  $(0, t_1]$ .
- Second largest *capacity*  $M_2$  comes from times  $(t_1, t_2]$ .

“Capacities decrease as the corresponding times  $\tau_i$  increase”

But this rule is not an operating requirement, it is optional...

# CAPACITY SAMPLING PLAN



- For highly reliable networks most trajectories exceed  $T$  in  $t^* < 1$ .
- For all such trajectories there is some work to be done and certain amount of splitting to perform.
- In order to reduce this work and, therefore, the simulation time, it is convenient that these trajectories exceed  $T$  *as soon as possible*.
- To achieve (or at least to get close to) this objective, larger capacities should be sampled earlier.