

The Cross-Entropy Method for Network Reliability Estimation

Dirk P. Kroese[◇]

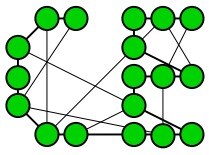
[◇]Department of Mathematics, University of Queensland, Brisbane

Kin-Ping Hui, Miro Kraetzl,

Defence Science and Technology Organisation

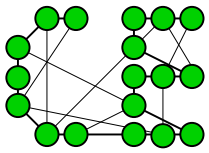
Nigel Bean

Department of Applied Mathematics, University of Adelaide



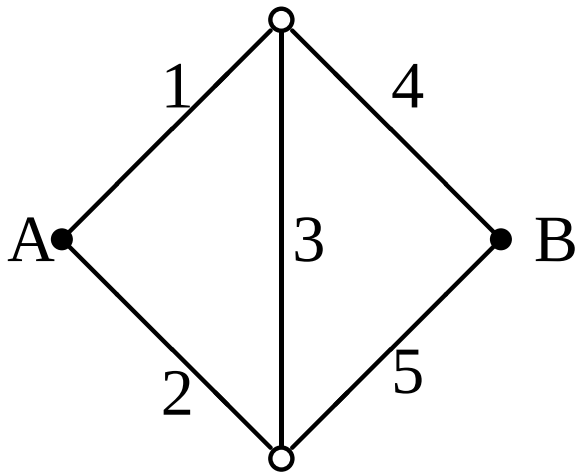
Contents

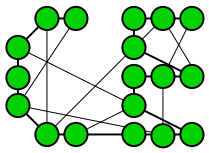
1. Network Reliability
2. Estimation of Network Reliability
 - Crude Monte Carlo + CE
 - Permutation Monte Carlo + CE
 - Merge Process
3. Numerical Experiments
4. Conclusions



Example: Bridge Network

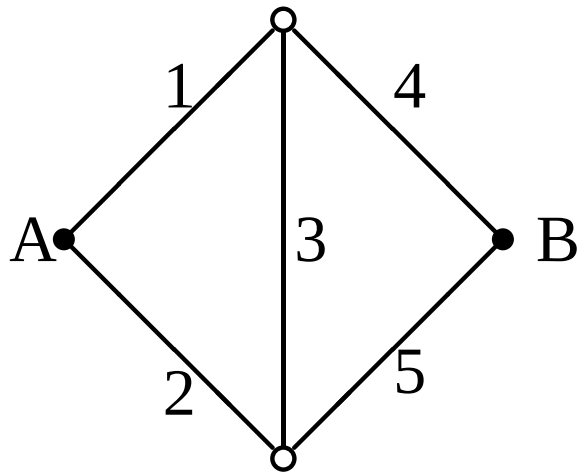
Network with 5 unreliable edges; works if the terminal nodes A and B are connected by operational edges.



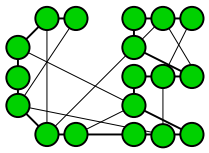


Example: Bridge Network

Network with 5 unreliable edges; works if the terminal nodes A and B are **connected** by operational edges.

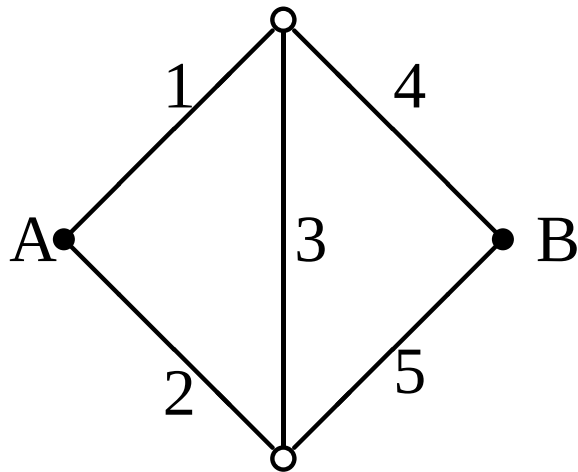


- X_e : edge state (1 = works, 0 = failed).
- $p_e = \mathbb{P}[X_e = 1]$: link reliability.
- Links fail **independently**



Example: Bridge Network

Network with 5 unreliable edges; works if the terminal nodes A and B are **connected** by operational edges.

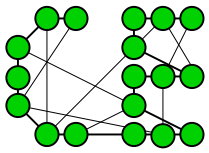


■ X_e : edge state (1 = works, 0 = failed).

■ $p_e = \mathbb{P}[X_e = 1]$: link reliability.

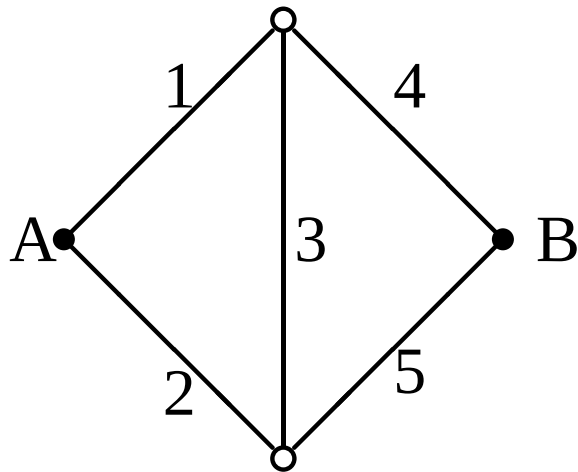
■ Links fail **independently**

$$\text{System state: } \varphi(\mathbf{X}) = \begin{cases} 1 & \text{if } A \text{ and } B \text{ connected,} \\ 0 & \text{otherwise.} \end{cases}$$



Example: Bridge Network

Network with 5 unreliable edges; works if the terminal nodes A and B are **connected** by operational edges.



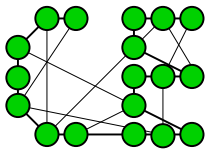
■ X_e : edge state (1 = works, 0 = failed).

■ $p_e = \mathbb{P}[X_e = 1]$: link reliability.

■ Links fail **independently**

System **state**: $\varphi(\mathbf{X}) = \begin{cases} 1 & \text{if } A \text{ and } B \text{ connected,} \\ 0 & \text{otherwise.} \end{cases}$

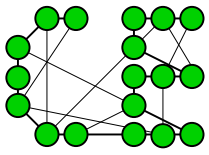
System **reliability**: $r = \mathbb{P}[\varphi(\mathbf{X}) = 1]$.



Bridge Network (cont.)

In this case the **structure function** φ is given by

$$\varphi(\mathbf{x}) = 1 - (1 - x_1 x_3 x_5)(1 - x_2 x_3 x_4)(1 - x_1 x_4)(1 - x_2 x_5) .$$



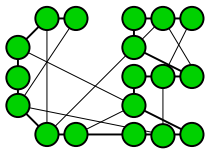
Bridge Network (cont.)

In this case the structure function φ is given by

$$\varphi(\mathbf{x}) = 1 - (1 - x_1x_3x_5)(1 - x_2x_3x_4)(1 - x_1x_4)(1 - x_2x_5) .$$

For highly reliable networks it is more useful to consider the unreliability

$$\bar{r} = 1 - r .$$



Bridge Network (cont.)

In this case the **structure function** φ is given by

$$\varphi(\mathbf{x}) = 1 - (1 - x_1 x_3 x_5)(1 - x_2 x_3 x_4)(1 - x_1 x_4)(1 - x_2 x_5) .$$

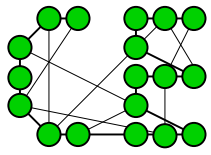
For **highly reliable** networks it is more useful to consider the **unreliability**

$$\bar{r} = 1 - r .$$

The system **unreliability** is equal to

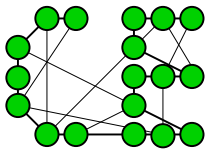
$$\begin{aligned} \bar{r} = & q_1 q_2 + q_2 q_3 q_4 - q_1 q_2 q_3 q_4 + q_1 q_3 q_5 - q_1 q_2 q_3 q_5 \\ & + q_4 q_5 - q_1 q_2 q_4 q_5 - q_1 q_3 q_4 q_5 - q_2 q_3 q_4 q_5 + 2 q_1 q_2 q_3 q_4 q_5 , \end{aligned}$$

where $\{q_i\}$ are the **link unreliabilities**.



Crude Monte Carlo Simulation

We want to estimate $\bar{r} = \mathbb{P}[\varphi(\mathbf{X}) = 0]$.



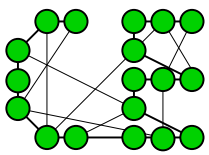
Crude Monte Carlo Simulation

We want to estimate $\bar{r} = \mathbb{P}[\varphi(\mathbf{X}) = 0]$.

Let $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \sim \mathbf{X}$ be iid random vectors. Then,

$$Q = \frac{1}{N} \sum_{i=1}^N (1 - \varphi(\mathbf{X}^{(i)}))$$

is an unbiased estimator for $\bar{r}$



Crude Monte Carlo Simulation

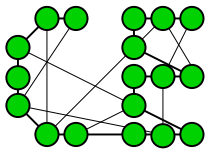
We want to estimate $\bar{r} = \mathbb{P}[\varphi(\mathbf{X}) = 0]$.

Let $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \sim \mathbf{X}$ be iid random vectors. Then,

$$Q = \frac{1}{N} \sum_{i=1}^N (1 - \varphi(\mathbf{X}^{(i)}))$$

is an unbiased estimator for $\bar{r}$ and its relative error is

$$\text{re}(Q) = \sqrt{\frac{\text{Var}(Q)}{(\mathbb{E}[Q])^2}} = \sqrt{\frac{1 - \bar{r}}{N \bar{r}}}.$$



Crude Monte Carlo Simulation

We want to estimate $\bar{r} = \mathbb{P}[\varphi(\mathbf{X}) = 0]$.

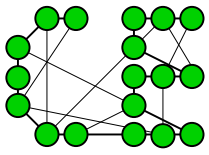
Let $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \sim \mathbf{X}$ be iid random vectors. Then,

$$Q = \frac{1}{N} \sum_{i=1}^N (1 - \varphi(\mathbf{X}^{(i)}))$$

is an unbiased estimator for $\bar{r}$ and its relative error is

$$\text{re}(Q) = \sqrt{\frac{\text{Var}(Q)}{(\mathbb{E}[Q])^2}} = \sqrt{\frac{1 - \bar{r}}{N \bar{r}}}.$$

For small $\bar{r}$, a large sample size N is needed.



CE and CMC

Let Y_i be the i -th **repair time** in the CP. Thus,

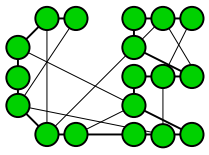
$$Y_i \sim \text{Exp}(\lambda_i) \quad \text{with} \quad \lambda_i = -\log q_i.$$

Note: $\mathbb{P}[Y_i \geq 1] = e^{-\lambda_i} = q_i$.

Let $S(\mathbf{Y})$ be the time at which the **system is repaired**. Then, we can write

$$\bar{r} = \mathbb{P}[S(\mathbf{Y}) \geq 1] .$$

Hence, we have written the estimation of $\bar{r}$ in the “standard” rare-event simulation form and can directly apply the **CE method for rare event simulation**.



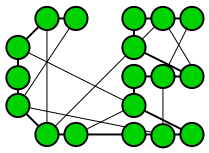
CE Algorithm

1. Let $\mathbf{v}_0 = \mathbf{u}_0$ (parameterization via mean). Set $t = 1$.
2. Sample $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ from the Exp distr. with mean $\mathbf{v}_{t-1}$.
3. Let $\mathcal{I}$ be the indices of the ρN largest repair times. Let γ_t be the smallest of these repair times.
4. Update $v_{t,j} := \frac{\sum_{i \in \mathcal{I}} W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_{t-1}) Y_{i,j}}{\sum_{i \in \mathcal{I}} W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_{t-1})}$,

where $W(\mathbf{y}; \mathbf{u}, \mathbf{v}) = f(\mathbf{y}; \mathbf{u})/f(\mathbf{y}; \mathbf{v})$ is the LR.

5. After $\gamma_T > 1$, estimate $\bar{r}$ via importance sampling, as

$$\hat{\bar{r}} = \frac{1}{N_1} \sum_{i=1}^{N_1} I\{S(\mathbf{Y}_i) \geq 1\} W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_T) .$$



Numerical Experiments

Bridge Network

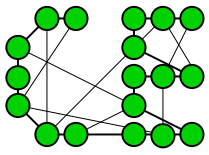
Nominal parameter: $\mathbf{u} = (0.3, 0.1, 0.8, 0.1, 0.2)$.

Parameters: $N = 2000$, $\rho = 0.01$. Final sample size 10^6 .

Typical result of the simulations:

	$\hat{r}$	$\hat{r}e$	$\widehat{\text{var}}$
CE-CMC	6.99111e-05	0.0166777	1.35945e-12
CMC	6.1e-05	0.128033	6.09963e-11
True value	7.07868e-05		

With minimal effort 98% reduction in RE (50-fold speedup).

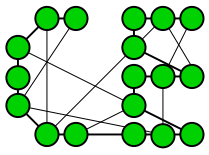


Evolution of CE algorithm

The algorithm stopped after two iterations.

t	$\hat{\gamma}_t$	$\hat{\mathbf{v}}_t$				
0	—	0.3	0.1	0.8	0.1	0.2
1	0.507	0.964833	0.216927	1.20908	0.0892952	0.567551
2	1.000	1.19792	0.120166	1.57409	0.0630103	1.15137

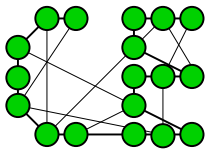
Notice that the algorithm tilts the parameters of the [mincut](#) elements $\{1, 3, 5\}$ to higher values while leaving the rest unimportant.



A Related Dynamic Network

Consider the following **dynamic** network:

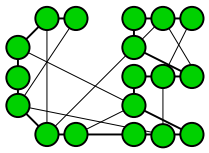
- At time $t = 0$ all edges are **failed**.



A Related Dynamic Network

Consider the following **dynamic** network:

- At time $t = 0$ all edges are **failed**.
- Each edge e has an exponential **repair** time with repair rate $\lambda(e) = -\log(q_e)$.

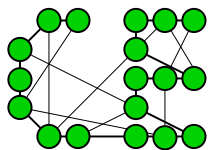


A Related Dynamic Network

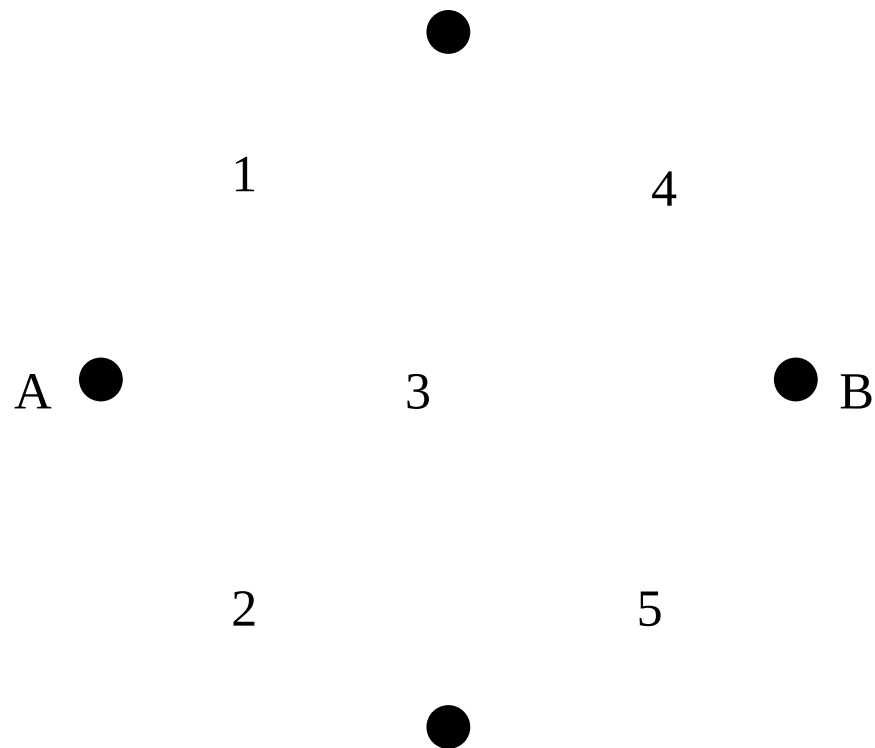
Consider the following **dynamic** network:

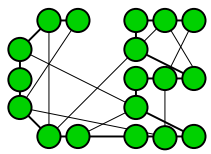
- At time $t = 0$ all edges are **failed**.
- Each edge e has an exponential **repair** time with repair rate $\lambda(e) = -\log(q_e)$.

Then $(\mathbf{X}(t))$ is a **Markov** process with state space $\{0, 1\}^m$. This process is called the **Construction Process** (CP) of the network.

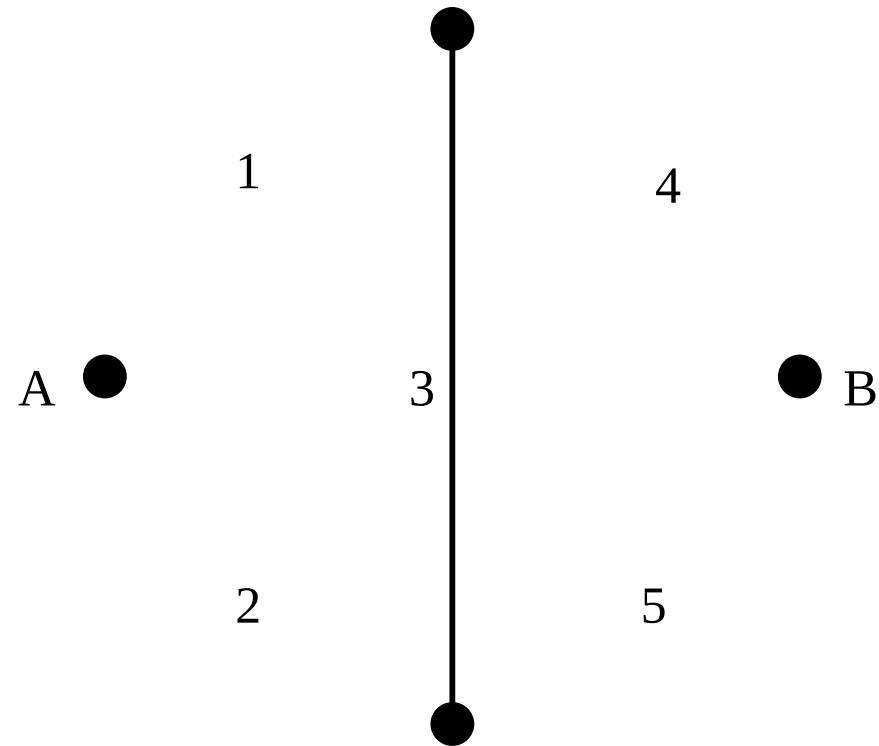


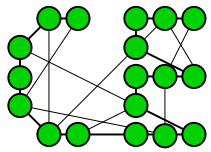
Example



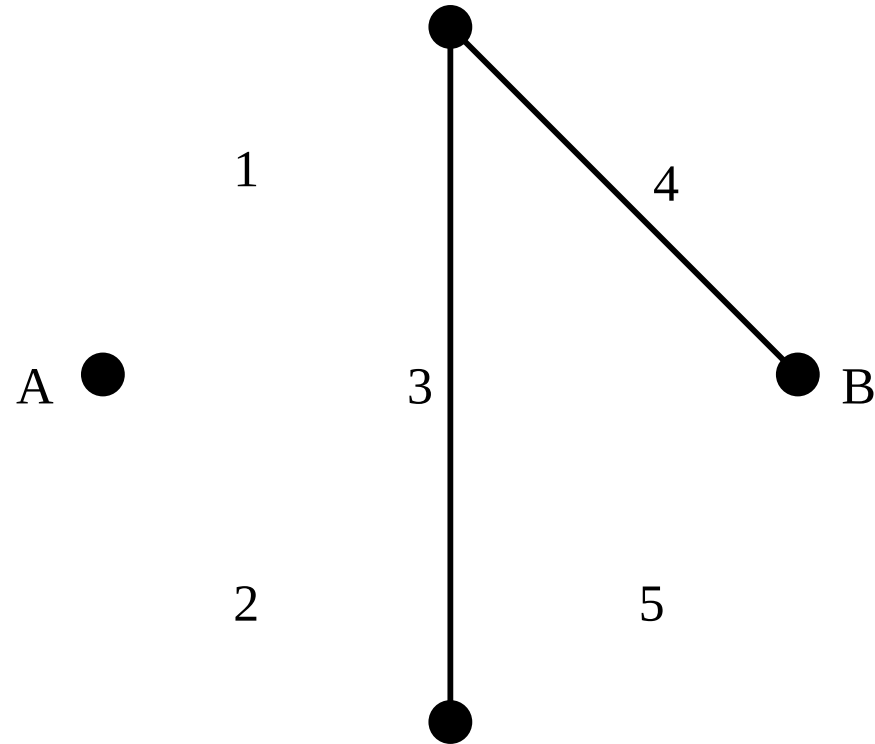


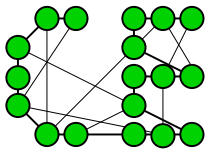
Example



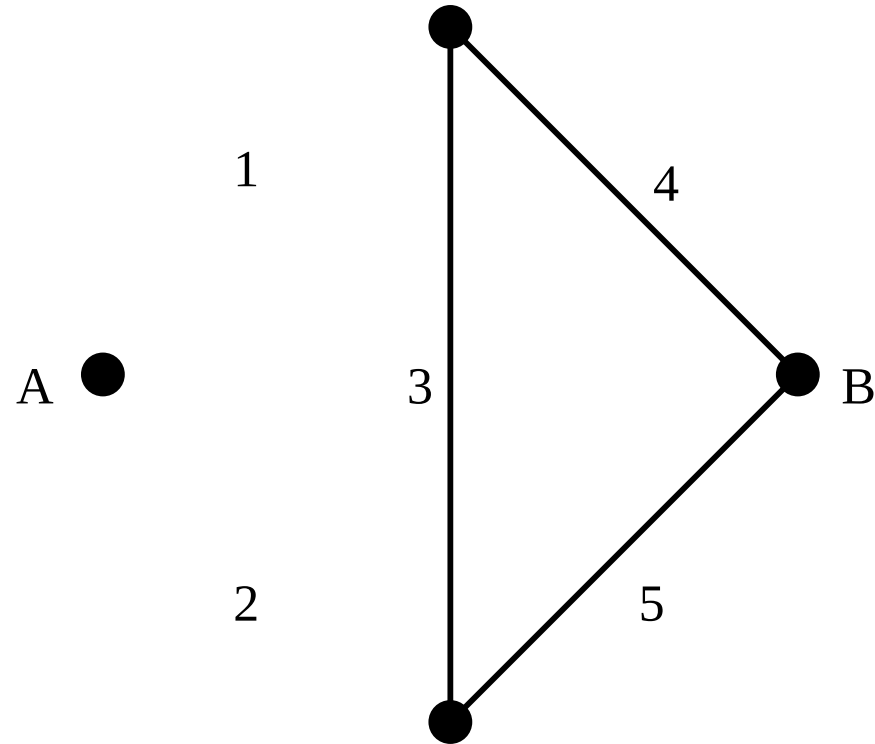


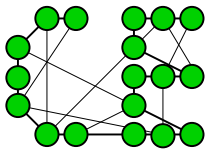
Example



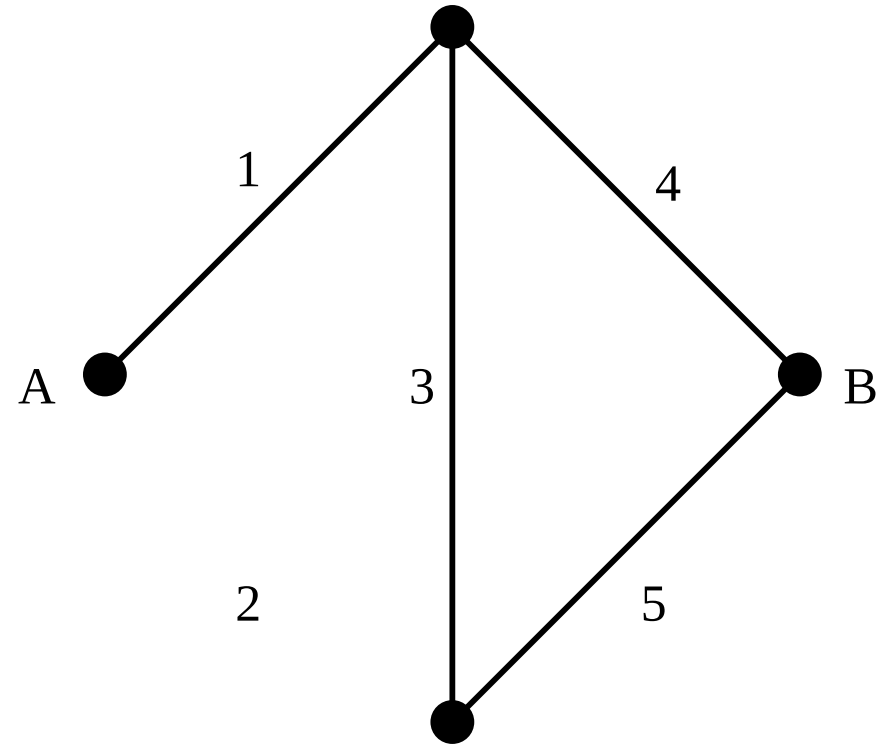


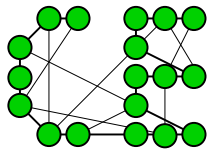
Example



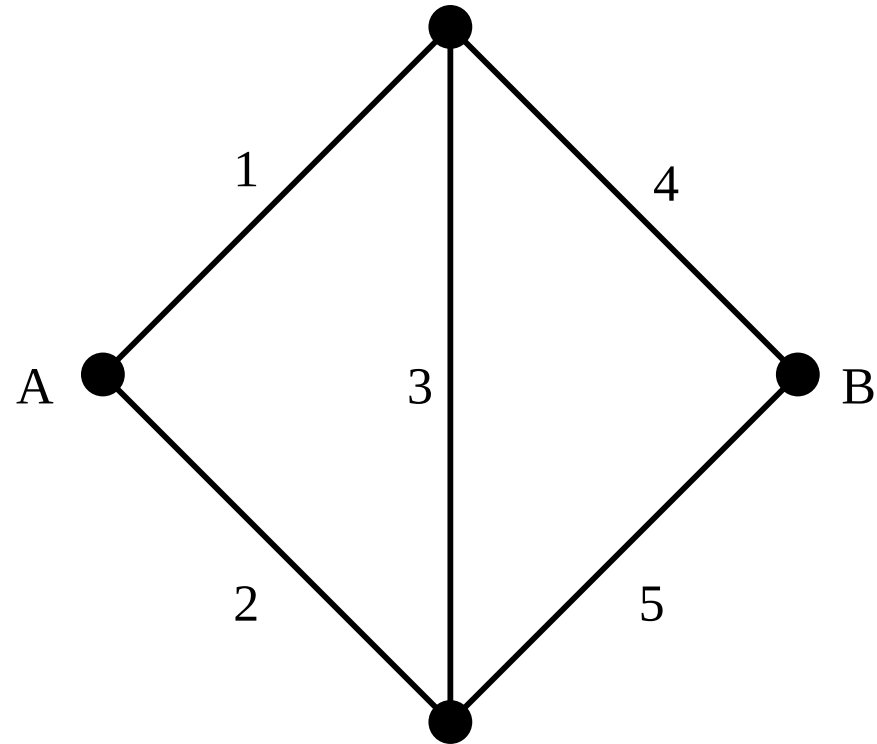


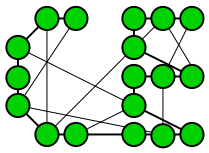
Example



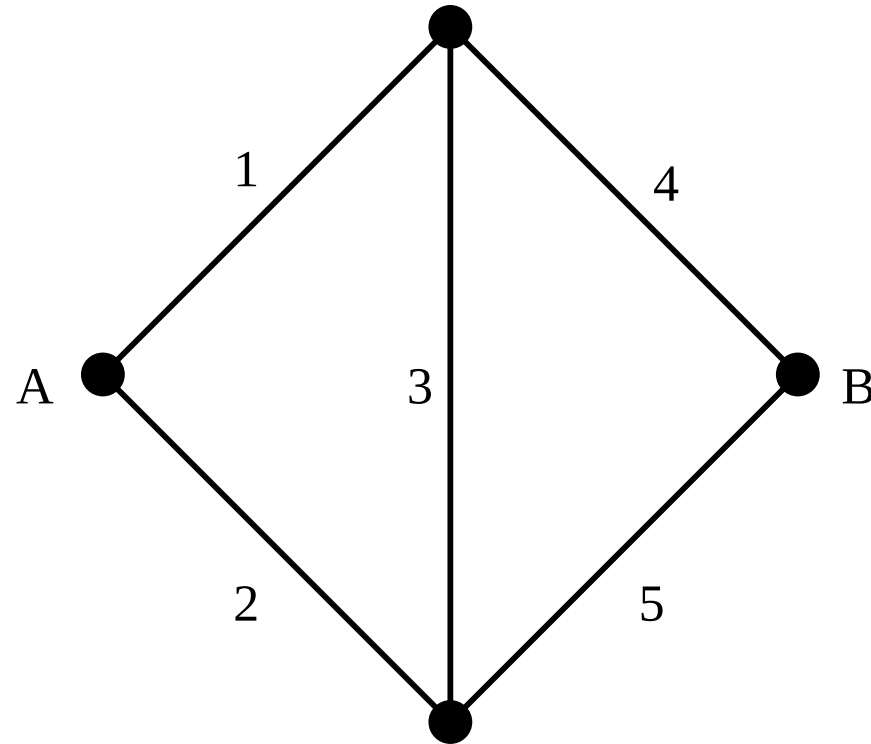


Example

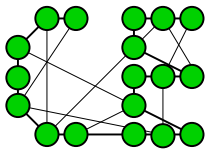




Example

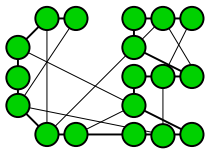


Order in which edges came up: $\pi = (3, 4, 5, 1, 2)$.
System becomes functional after $b(\pi) = 4$ steps
(critical number).



Permutation Monte Carlo

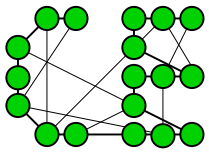
Let Π denote the **order** (permutation) in which the edges become operational.



Permutation Monte Carlo

Let Π denote the **order** (permutation) in which the edges become operational.

Let $A_0, A_1, A_2, \dots$ be the corresponding **sojourn times** of $(\mathbf{X}(t))$.



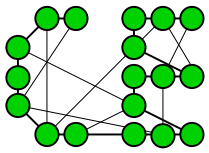
Permutation Monte Carlo

Let Π denote the **order** (permutation) in which the edges become operational.

Let $A_0, A_1, A_2, \dots$ be the corresponding **sojourn times** of $(\mathbf{X}(t))$.

Let $E_0, E_1, \dots$ denote the **sets of failed edges**, and let

$$\lambda(E_i) = \sum_{e \in E_i} \lambda(e),$$



Permutation Monte Carlo

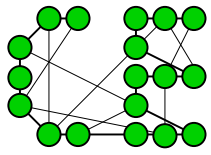
Let Π denote the **order** (permutation) in which the edges become operational.

Let $A_0, A_1, A_2, \dots$ be the corresponding **sojourn times** of $(\mathbf{X}(t))$.

Let $E_0, E_1, \dots$ denote the **sets of failed edges**, and let

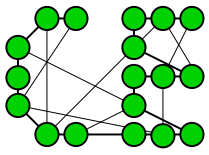
$$\lambda(E_i) = \sum_{e \in E_i} \lambda(e),$$

Conditional on $\{\Pi = \pi\}$, $A_0, \dots, A_{m-1}$ are **independent**, and $A_i \sim \text{Exp}(\lambda(E_i))$.



Permutation Monte Carlo

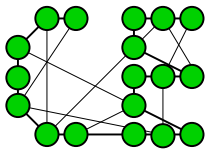
Since $\mathbb{P}[X_e(1) = 1] = p_e$, we have:



Permutation Monte Carlo

Since $\mathbb{P}[X_e(1) = 1] = p_e$, we have:

At time 1, reliability of dynamic netw. = reliability of original netw.



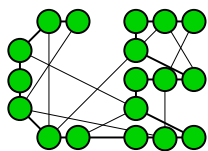
Permutation Monte Carlo

Since $\mathbb{P}[X_e(1) = 1] = p_e$, we have:

At time 1, reliability of **dynamic** netw. = reliability of **original** netw.

Thus,

$$\begin{aligned}\bar{r} &= \sum_{\pi} \mathbb{P}[\varphi(\mathbf{X}(1)) = 0 \mid \Pi = \pi] \mathbb{P}[\Pi = \pi] \\ &= \sum_{\pi} \mathbb{P}[A_0 + \cdots + A_{b(\pi)-1} > 1 \mid \Pi = \pi] \mathbb{P}[\Pi = \pi] = \mathbb{E}[G(\Pi)] .\end{aligned}$$



Permutation Monte Carlo

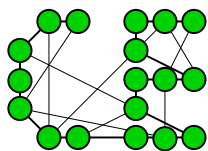
Since $\mathbb{P}[X_e(1) = 1] = p_e$, we have:

At time 1, reliability of **dynamic** netw. = reliability of **original** netw.

Thus,

$$\begin{aligned}\bar{r} &= \sum_{\pi} \mathbb{P}[\varphi(\mathbf{X}(1)) = 0 \mid \Pi = \pi] \mathbb{P}[\Pi = \pi] \\ &= \sum_{\pi} \mathbb{P}[A_0 + \cdots + A_{b(\pi)-1} > 1 \mid \Pi = \pi] \mathbb{P}[\Pi = \pi] = \mathbb{E}[G(\Pi)] .\end{aligned}$$

$$\hat{\bar{r}} = \frac{1}{N} \sum_{i=1}^N G(\Pi_{(i)}) \quad \text{is an unbiased estimator for } \bar{r}.$$



CE and PMC

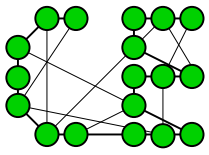
Thus: $\bar{r} = \mathbb{E}[G(\Pi)]$, where

- G is a known function (convolution),
- Π can be viewed as a function of $\mathbf{Y}$.

Hence,

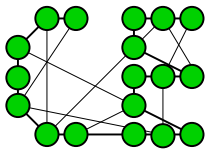
$$\bar{r} = \mathbb{E}_{\mathbf{u}}[G(\Pi(\mathbf{Y}))] := \mathbb{E}_{\mathbf{u}}[S(\mathbf{Y})] = \mathbb{E}_{\mathbf{v}}[S(\mathbf{Y}) W(\mathbf{Y}; \mathbf{u}, \mathbf{v})] .$$

This suggests estimating $\bar{r}$ via IS, letting CE find the optimal parameter $\mathbf{v}$.



Updating

In this case we have a **one-phase** CE procedure in which we only update v_t .

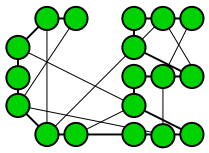


Updating

In this case we have a **one-phase** CE procedure in which we only update $\mathbf{v}_t$.

The **updating formula** for the i -th component of $\mathbf{v}_t$ is

$$v_{t,j} = \frac{\sum_{i=1}^N S(\mathbf{Y}_i) W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_{t-1}) Y_{i,j}}{\sum_{i=1}^N S(\mathbf{Y}_i) W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_{t-1})},$$



Updating

In this case we have a **one-phase** CE procedure in which we only update $\mathbf{v}_t$.

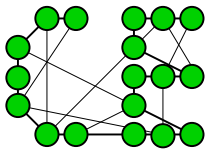
The **updating formula** for the i -th component of $\mathbf{v}_t$ is

$$v_{t,j} = \frac{\sum_{i=1}^N S(\mathbf{Y}_i) W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_{t-1}) Y_{i,j}}{\sum_{i=1}^N S(\mathbf{Y}_i) W(\mathbf{Y}_i; \mathbf{u}, \mathbf{v}_{t-1})},$$

To improve convergence it is beneficial to use a **smoothing procedure**. That is we take

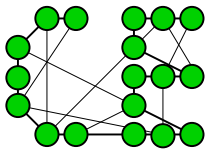
$$\mathbf{v}'_t = \alpha \mathbf{v}_t + (1 - \alpha) \mathbf{v}_{t-1}$$

as the new parameter vector for the next iteration.

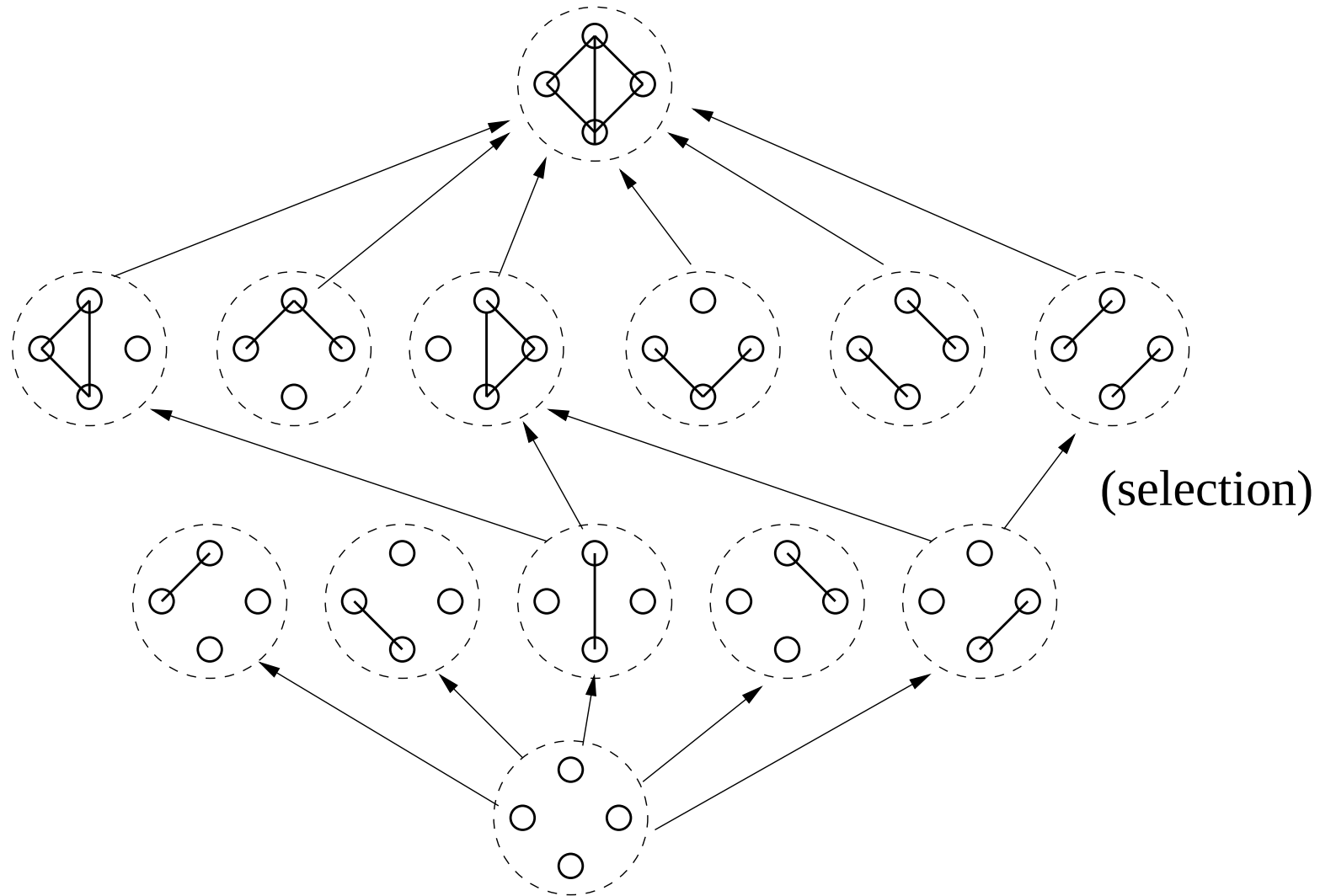


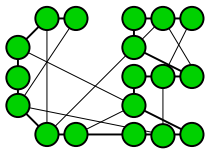
Merge Process

A closer look at the evolution of the Construction Process reveals that many of the above results remain valid when we **merge** various states into “**super states**” at various stages of the process. This is known as the **Merge Process** (MP).



States of MP (Bridge Example)



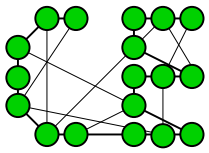


Numerical Results (bridge network)

	$\hat{r}$	$\hat{r}e$	$\widehat{\text{var}}$
CE-MP	7.08223e-05	0.00116362	6.79140e-15
MP	7.08109e-05	0.00132036	8.74143e-15
CE-PMC	7.07905e-05	0.00120961	7.33236e-15
PMC	7.07850e-05	0.00132445	8.78930e-15
CE-CMC	6.99111e-05	0.01667770	1.35945e-12
True value	7.07868e-05		

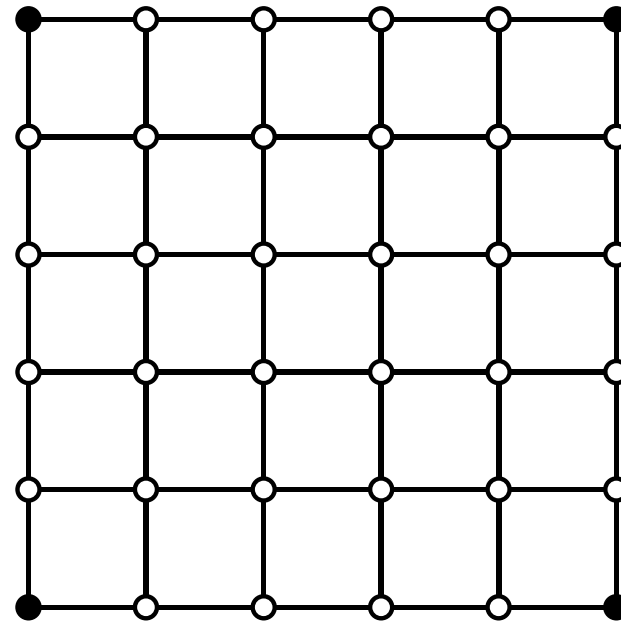
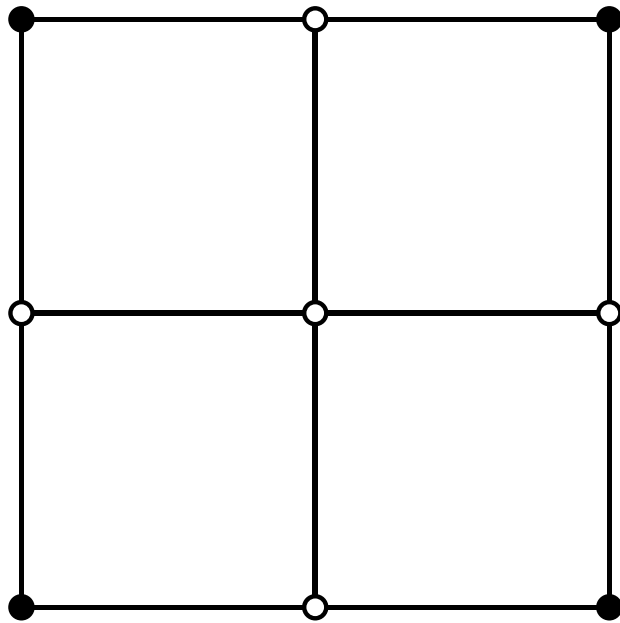
CE-CMC has 10 times the RE of the other methods.

CE consistently gives an improvement.

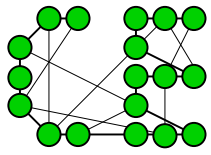


Larger examples

Grid networks. Terminal nodes (black) to be connected. All links have the **same unreliability** q .

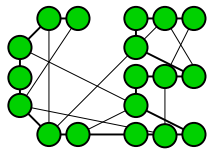


In all the experiments we use $N = 5000$ (for CE) and a final sample size of 10^6 .



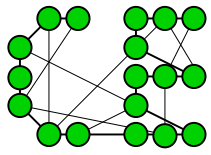
3×3 grid, $q = 10^{-3}$

	T	α	$\hat{r}$	$\hat{r}e$	$\widehat{\text{var}}$
CE-MP	10	0.1	4.01172e-06	0.001071	1.84515e-17
MP	-	-	4.01179e-06	0.001122	2.02652e-17
CE-PMC	10	0.1	4.01057e-06	0.003420	1.88147e-16
PMC	-	-	4.01609e-06	0.003895	2.44653e-16
CE-CMC	4	1	2.8296e-06	0.150517	1.81393e-13
CMC	-	-	4e-06	0.499999	3.99998e-12
True value			4.01199e-06		



3×3 grid, $q = 10^{-6}$

	T	α	$\hat{r}$	$\hat{r}_e$	$\widehat{\text{var}}$
CE-MP	10	0.1	3.99876e-12	0.001076	1.85116e-29
MP	-	-	4.00279e-12	0.001127	2.03639e-29
CE-PMC	10	0.1	4.00797e-12	0.003455	1.91801e-28
PMC	-	-	3.99354e-12	0.003940	2.47621e-28
CE-CMC	5	1	1.75878e-12	0.176095	9.59220e-26
CMC	-	-	0	-	-
True value			4.00001e-12.		

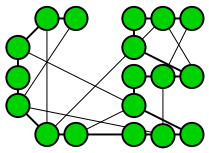


6×6 grid, $q = 10^{-3}$

	T	α	$\hat{r}$	$\hat{re}$	$\widehat{var}$
CE-MP	10	0.1	4.00239e-06	0.001528	3.74067e-17
MP	-	-	4.01041e-06	0.001745	4.89923e-17
CE-PMC	10	0.1	4.02088e-06	0.011778	2.24284e-15
PMC	-	-	3.95377e-06	0.020306	6.44547e-15
CE-CMC	4	1	1.32219e-06	0.489052	4.18118e-13
CMC	-	-	6e-06	0.408247	5.99996e-12
True value	4.00800e-06.				

CE reduces the RE of MP by 25%. Even more for PMC.

Similar findings for $q = 10^{-6}$.



Conclusions and Future Research

- CE gives substantial improvement over the CMC estimation of network reliability.
- PMC much better than CMC and CE-CMC (Conditional Sampling Reduced Variance!)
- CE-PMC can still improve on PMC (5-50%).
- Unreliable nodes
- Network planning. Reliability optimization. Noisy optimization.