STK3405 – Lecture 9

A. B. Huseby & K. R. Dahl

Department of Mathematics
University of Oslo, Norway

Chapter 7

Computing reliability using conditional Monte Carlo methods

Section 7.1

Monte Carlo simulation and conditioning

Monte Carlo simulation and conditioning

Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be a random vector with a known distribution, and let $\phi = \phi(\boldsymbol{X})$. The distribution of ϕ cannot be derived analytically in polynomial time with respect to n. We want to estimate:

$$h = E[\phi(\boldsymbol{X})]$$

We run a Monte Carlo simulation generating N independent vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$, all having the same distribution as $\boldsymbol{X}$, and estimate h by the simple *Monte Carlo* estimate:

$$\hat{h}_{MC} = \frac{1}{N} \sum_{r=1}^{N} \phi(\boldsymbol{X}_r).$$

Then $E[\hat{h}_{MC}] = h$, and we have:

$$\mathrm{Var}\left(\hat{h}_{MC}\right) = \frac{1}{N^2} \sum_{r=1}^{N} \mathrm{Var}(\phi) = \frac{1}{N} \mathrm{Var}(\phi)$$

Monte Carlo simulation and conditioning

Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be a random vector with a known distribution, and let $\phi = \phi(\boldsymbol{X})$. The distribution of ϕ cannot be derived analytically in polynomial time with respect to n. We want to estimate:

$$h = E[\phi(\boldsymbol{X})]$$

We run a Monte Carlo simulation generating N independent vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$, all having the same distribution as $\boldsymbol{X}$, and estimate h by the simple *Monte Carlo* estimate:

$$\hat{h}_{MC} = \frac{1}{N} \sum_{r=1}^{N} \phi(\boldsymbol{X}_r).$$

Then $E[\hat{h}_{MC}] = h$, and we have:

$$\mathrm{Var}\left(\hat{h}_{MC}\right) = \frac{1}{N^2} \sum_{r=1}^{N} \mathrm{Var}(\phi) = \frac{1}{N} \mathrm{Var}(\phi)$$

Monte Carlo simulation and conditioning

Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be a random vector with a known distribution, and let $\phi = \phi(\boldsymbol{X})$. The distribution of ϕ cannot be derived analytically in polynomial time with respect to n. We want to estimate:

$$h = E[\phi(\boldsymbol{X})]$$

We run a Monte Carlo simulation generating N independent vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$, all having the same distribution as $\boldsymbol{X}$, and estimate h by the simple *Monte Carlo* estimate:

$$\hat{h}_{MC} = \frac{1}{N} \sum_{r=1}^{N} \phi(\boldsymbol{X}_r).$$

Then $E[\hat{h}_{MC}] = h$, and we have:

$$\mathrm{Var}\left(\hat{h}_{MC}\right) = \frac{1}{N^2} \sum_{r=1}^{N} \mathrm{Var}(\phi) = \frac{1}{N} \mathrm{Var}(\phi)$$

Monte Carlo simulation and conditioning (cont.)

Let $S = S(\boldsymbol{X})$ is a *discrete* variable with values in the set $\{s_1, \ldots, s_k\}$, and with a distribution that can be calculated analytically in polynomial time with respect to n.

We also introduce:

$$\theta_j = E[\phi \mid S = s_j], \quad j = 1, \ldots, k.$$

We then have:

$$h = E[\phi] = \sum_{j=1}^{k} E[\phi \mid S = s_j] P(S = s_j) = \sum_{j=1}^{k} \theta_j P(S = s_j)$$

Monte Carlo simulation and conditioning (cont.)

Let $S = S(\boldsymbol{X})$ is a *discrete* variable with values in the set $\{s_1, \ldots, s_k\}$, and with a distribution that can be calculated analytically in polynomial time with respect to n.

We also introduce:

$$\theta_j = E[\phi \mid S = s_j], \quad j = 1, \ldots, k.$$

We then have:

$$h = E[\phi] = \sum_{j=1}^{k} E[\phi \mid S = s_j] P(S = s_j) = \sum_{j=1}^{k} \theta_j P(S = s_j)$$

Monte Carlo simulation and conditioning (cont.)

Let $S = S(\boldsymbol{X})$ is a *discrete* variable with values in the set $\{s_1, \ldots, s_k\}$, and with a distribution that can be calculated analytically in polynomial time with respect to n.

We also introduce:

$$\theta_j = E[\phi \mid S = s_j], \quad j = 1, \ldots, k.$$

We then have:

$$h = E[\phi] = \sum_{j=1}^{k} E[\phi \mid S = s_j] P(S = s_j) = \sum_{j=1}^{k} \theta_j P(S = s_j)$$

Monte Carlo simulation and conditioning (cont.)

Instead of generating N samples from the distribution of $\boldsymbol{X}$, we divide the sample set into k groups, one for each possible value of S.

The j-th group has size N_j, $j = 1, \ldots, k$, and $N_1 + \cdots + N_k = N$.

The samples in the j-th group, $\boldsymbol{X}_{1,j}, \ldots, \boldsymbol{X}_{N_j,j}$, are sampled from the conditional distribution of $\boldsymbol{X}$ given that $S = s_j$, $j = 1, \ldots, k$.

Then θ_j is estimated by:

$$\hat{\theta}_j = \frac{1}{N_j} \sum_{r=1}^{N_j} \phi(\boldsymbol{X}_{r,j}), \quad j = 1, \ldots, k.$$

Monte Carlo simulation and conditioning (cont.)

Instead of generating N samples from the distribution of $\boldsymbol{X}$, we divide the sample set into k groups, one for each possible value of S.

The j-th group has size N_j, $j = 1, \ldots, k$, and $N_1 + \cdots + N_k = N$.

The samples in the j-th group, $\boldsymbol{X}_{1,j}, \ldots, \boldsymbol{X}_{N_j,j}$, are sampled from the conditional distribution of $\boldsymbol{X}$ given that $S = s_j$, $j = 1, \ldots, k$.

Then θ_j is estimated by:

$$\hat{\theta}_j = \frac{1}{N_j} \sum_{r=1}^{N_j} \phi(\boldsymbol{X}_{r,j}), \quad j = 1, \ldots, k.$$

Monte Carlo simulation and conditioning (cont.)

Instead of generating N samples from the distribution of $\boldsymbol{X}$, we divide the sample set into k groups, one for each possible value of S.

The j-th group has size N_j, $j = 1, \ldots, k$, and $N_1 + \cdots + N_k = N$.

The samples in the j-th group, $\boldsymbol{X}_{1,j}, \ldots, \boldsymbol{X}_{N_j,j}$, are sampled from the conditional distribution of $\boldsymbol{X}$ given that $S = s_j$, $j = 1, \ldots, k$.

Then θ_j is estimated by:

$$\hat{\theta}_j = \frac{1}{N_j} \sum_{r=1}^{N_j} \phi(\boldsymbol{X}_{r,j}), \quad j = 1, \ldots, k.$$

Monte Carlo simulation and conditioning (cont.)

We then have:

$$E[\hat{\theta}_j] = \frac{1}{N_j}\sum_{r=1}^{N_j} E[\phi(\boldsymbol{X}_{r,j})] = \frac{1}{N_j}\sum_{r=1}^{N_j} E[\phi|S = s_j]$$
$$= \frac{1}{N_j}\sum_{r=1}^{N_j} \theta_j = \theta_j, \qquad j = 1, \ldots, k.$$

Moreover, the variances of these estimates are:

$$\mathrm{Var}(\hat{\theta}_j) = \frac{1}{N_j^2}\sum_{r=1}^{N_j} \mathrm{Var}(\phi(\boldsymbol{X}_{r,j})) = \frac{1}{N_j^2}\sum_{r=1}^{N_j} \mathrm{Var}(\phi|S = s_j)$$
$$= \frac{1}{N_j}\mathrm{Var}(\phi|S = s_j), \qquad j = 1, \ldots, k.$$

Monte Carlo simulation and conditioning (cont.)

We then have:

$$E[\hat{\theta}_j] = \frac{1}{N_j}\sum_{r=1}^{N_j} E[\phi(\boldsymbol{X}_{r,j})] = \frac{1}{N_j}\sum_{r=1}^{N_j} E[\phi | S = s_j]$$
$$= \frac{1}{N_j}\sum_{r=1}^{N_j} \theta_j = \theta_j, \qquad j = 1, \ldots, k.$$

Moreover, the variances of these estimates are:

$$\mathrm{Var}(\hat{\theta}_j) = \frac{1}{N_j^2}\sum_{r=1}^{N_j} \mathrm{Var}(\phi(\boldsymbol{X}_{r,j})) = \frac{1}{N_j^2}\sum_{r=1}^{N_j} \mathrm{Var}(\phi | S = s_j)$$
$$= \frac{1}{N_j}\mathrm{Var}(\phi | S = s_j), \qquad j = 1, \ldots, k.$$

Monte Carlo simulation and conditioning (cont.)

By combining $\hat{\theta}_1, \ldots, \hat{\theta}_k$, we get the *conditional Monte Carlo estimate*:

$$\hat{h}_{CMC} = \sum_{j=1}^{k} \hat{\theta}_j P(S = s_j).$$

Since $\hat{\theta}_1, \ldots, \hat{\theta}_k$ are independent, the variance of the conditional Monte Carlo estimate is:

$$\mathrm{Var}(\hat{h}_{CMC}) = \mathrm{Var}[\sum_{j=1}^{k} \hat{\theta}_j P(S = s_j)] = \sum_{j=1}^{k} \mathrm{Var}(\hat{\theta}_j)[P(S = s_j)]^2$$

$$= \sum_{j=1}^{k} \frac{1}{N_j} \mathrm{Var}(\phi | S = s_j)[P(S = s_j)]^2.$$

Monte Carlo simulation and conditioning (cont.)

By combining $\hat{\theta}_1, \ldots, \hat{\theta}_k$, we get the *conditional Monte Carlo estimate*:

$$\hat{h}_{CMC} = \sum_{j=1}^{k} \hat{\theta}_j P(S = s_j).$$

Since $\hat{\theta}_1, \ldots, \hat{\theta}_k$ are independent, the variance of the conditional Monte Carlo estimate is:

$$\begin{aligned}\mathrm{Var}(\hat{h}_{CMC}) &= \mathrm{Var}[\sum_{j=1}^{k} \hat{\theta}_j P(S = s_j)] = \sum_{j=1}^{k} \mathrm{Var}(\hat{\theta}_j)[P(S = s_j)]^2 \\ &= \sum_{j=1}^{k} \frac{1}{N_j} \mathrm{Var}(\phi | S = s_j)[P(S = s_j)]^2.\end{aligned}$$

Monte Carlo simulation and conditioning (cont.)

NOTE: The variance of $\hat{h}_{CMC}$ depends on the choices of the N_j's.

In order to compare the result with the variance of $\hat{h}_{MC}$, we let:

$$N_j \approx N \cdot P(S = s_j), \quad j = 1, \ldots, k.$$

Hence:

$$\sum_{j=1}^{k} N_j \approx \sum_{j=1}^{k} N \cdot P(S = s_j) = N \cdot \sum_{j=1}^{k} P(S = s_j) = N.$$

Monte Carlo simulation and conditioning (cont.)

With this choice we get:

$$\begin{aligned}
Var(\hat{h}_{CMC}) &\approx \sum_{j=1}^{k} \frac{1}{N \cdot P(S = s_j)} \mathrm{Var}(\phi | S = s_j)[P(S = s_j)]^2 \\
&= \frac{1}{N} \sum_{j=1}^{k} \mathrm{Var}(\phi | S = s_j) P(S = s_j) = \frac{1}{N} E[\mathrm{Var}(\phi | S)] \\
&= \frac{1}{N}\Big(\mathrm{Var}(\phi) - \mathrm{Var}[E(\phi | S)]\Big) \leq \frac{1}{N} \mathrm{Var}(\phi) = \mathrm{Var}(\hat{h}_{MC}).
\end{aligned}$$

Here we have used the well-known relation:

$$\mathrm{Var}(\phi) = \mathrm{Var}[E(\phi | S)] + E[\mathrm{Var}(\phi | S)]$$

Monte Carlo simulation and conditioning (cont.)

With this choice we get:

$$Var(\hat{h}_{CMC}) \approx \sum_{j=1}^{k} \frac{1}{N \cdot P(S = s_j)} \mathrm{Var}(\phi | S = s_j)[P(S = s_j)]^2$$

$$= \frac{1}{N} \sum_{j=1}^{k} \mathrm{Var}(\phi | S = s_j) P(S = s_j) = \frac{1}{N} E[\mathrm{Var}(\phi | S)]$$

$$= \frac{1}{N}\big(\mathrm{Var}(\phi) - \mathrm{Var}[E(\phi | S)]\big) \leq \frac{1}{N} \mathrm{Var}(\phi) = \mathrm{Var}(\hat{h}_{MC}).$$

Here we have used the well-known relation:

$$\mathrm{Var}(\phi) = \mathrm{Var}[E(\phi | S)] + E[\mathrm{Var}(\phi | S)]$$

Monte Carlo simulation and conditioning (cont.)

Since:

$$\begin{aligned}\mathrm{Var}(\hat{h}_{CMC}) &\approx \frac{1}{N}\big(\mathrm{Var}(\phi) - \mathrm{Var}[E(\phi|S)]\big)\\ &\leq \frac{1}{N}\mathrm{Var}(\phi) = \mathrm{Var}(\hat{h}_{MC}),\end{aligned}$$

it follows that $\mathrm{Var}(\hat{h}_{CMC}) < \mathrm{Var}(\hat{h}_{MC})$ whenever $\mathrm{Var}[E(\phi \mid S)] > 0$.

$\mathrm{Var}[E(\phi \mid S)]$ can be interpreted as a measure of how much information S contains relative to ϕ.

Monte Carlo simulation and conditioning (cont.)

1. S and ϕ independent. Then $E(\phi \mid S) = E(\phi)$, and thus,

$$\mathrm{Var}[E(\phi \mid S)] = 0,$$

implying that $\mathrm{Var}(\hat{h}_{CMC}) = \mathrm{Var}(\hat{h}_{MC})$.

2. $S = \phi$. Then $E(\phi \mid S) = \phi$, and thus,

$$\mathrm{Var}[E(\phi \mid S)] = \mathrm{Var}(\phi),$$

implying that $\mathrm{Var}(\hat{h}_{CMC}) = 0$.

3. $S = \boldsymbol{X}$. Then $E(\phi \mid S) = \phi(\boldsymbol{X})$, and thus,

$$\mathrm{Var}[E(\phi \mid S)] = \mathrm{Var}(\phi),$$

implying that $\mathrm{Var}(\hat{h}_{CMC}) = 0$.

Monte Carlo simulation and conditioning (cont.)

1. S and ϕ independent. Then $E(\phi \mid S) = E(\phi)$, and thus,

$$\text{Var}[E(\phi \mid S)] = 0,$$

implying that $\text{Var}(\hat{h}_{CMC}) = \text{Var}(\hat{h}_{MC})$.

2. $S = \phi$. Then $E(\phi \mid S) = \phi$, and thus,

$$\text{Var}[E(\phi \mid S)] = \text{Var}(\phi),$$

implying that $\text{Var}(\hat{h}_{CMC}) = 0$.

3. $S = \boldsymbol{X}$. Then $E(\phi \mid S) = \phi(\boldsymbol{X})$, and thus,

$$\text{Var}[E(\phi \mid S)] = \text{Var}(\phi),$$

implying that $\text{Var}(\hat{h}_{CMC}) = 0$.

Monte Carlo simulation and conditioning (cont.)

1. S and ϕ independent. Then $E(\phi \mid S) = E(\phi)$, and thus,
$$\mathrm{Var}[E(\phi \mid S)] = 0,$$
implying that $\mathrm{Var}(\hat{h}_{CMC}) = \mathrm{Var}(\hat{h}_{MC})$.

2. $S = \phi$. Then $E(\phi \mid S) = \phi$, and thus,
$$\mathrm{Var}[E(\phi \mid S)] = \mathrm{Var}(\phi),$$
implying that $\mathrm{Var}(\hat{h}_{CMC}) = 0$.

3. $S = \boldsymbol{X}$. Then $E(\phi \mid S) = \phi(\boldsymbol{X})$, and thus,
$$\mathrm{Var}[E(\phi \mid S)] = \mathrm{Var}(\phi),$$
implying that $\mathrm{Var}(\hat{h}_{CMC}) = 0$.

Monte Carlo simulation and conditioning (cont.)

In order to minimize the variance of the estimate, the variable S should contain as much information about ϕ as possible.

At the same time S must be chosen so that:

- The distribution of S can be derived analytically in polynomial time
- The number of possible values of S, i.e., k, must be bounded by a polynomial function of n
- It must be possible to sample efficiently from the distribution of $\boldsymbol{X}$ given S

Monte Carlo simulation and conditioning (cont.)

In order to minimize the variance of the estimate, the variable S should contain as much information about ϕ as possible.

At the same time S must be chosen so that:

- The distribution of S can be derived analytically in polynomial time
- The number of possible values of S, i.e., k, must be bounded by a polynomial function of n
- It must be possible to sample efficiently from the distribution of $\boldsymbol{X}$ given S

Monte Carlo simulation and conditioning (cont.)

In order to minimize the variance of the estimate, the variable S should contain as much information about ϕ as possible.

At the same time S must be chosen so that:

- The distribution of S can be derived analytically in polynomial time
- The number of possible values of S, i.e., k, must be bounded by a polynomial function of n
- It must be possible to sample efficiently from the distribution of $\boldsymbol{X}$ given S

Conditioning on the sum of the component state variables

Conditioning on the sum

Consider a binary monotone system (C, ϕ) where $C = \{1, \ldots, n\}$, and let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be the vector of the component state variables.

Moreover, let:

$$S = S(\boldsymbol{X}) = \sum_{i=1}^{n} X_i$$

Thus, the set of possible values for S is $\{0, 1, \ldots, n\}$, and we let:

$$\theta_s = E[\phi | S = s], \qquad s = 0, 1, \ldots, n.$$

We must find an efficient way of sampling from the conditional distribution of $\boldsymbol{X}$ given $S = s$, $s = 0, 1, \ldots, n$.

Conditioning on the sum (cont.)

This can be done as follows:

STEP 1. Sample X_1 from the conditional distribution of $X_1|S = s$, and let x_1 be the result.

STEP 2. Sample X_2 from the conditional distribution of $X_2|S = s, X_1 = x_1$, and let x_2 be the result.

...

STEP n. Sample X_n from the conditional distribution of $X_n|S = s, X_1 = x_1, \ldots X_{n-1} = x_{n-1}$, and let x_n be the result.

To compute the necessary conditional distributions we introduce the partial sums, $S_1, \ldots, S_n$:

$$S_m = \sum_{i=m}^{n} X_i, \quad m = 1, \ldots, n.$$

Conditioning on the sum (cont.)

We then have:

$$
\begin{aligned}
P(X_m &= x_m \mid X_1 = x_1, \ldots, X_{m-1} = x_{m-1}, S = s) \\
&= \frac{P(X_m = x_m, S = s \mid X_1 = x_1, \ldots, X_{m-1} = x_{m-1})}{P(S = s \mid X_1 = x_1, \ldots, X_{m-1} = x_{m-1})} \\
&= \frac{P(X_m = x_m, S_{m+1} = s - \sum_{j=1}^{m} x_j)}{P(S_m = s - \sum_{j=1}^{m-1} x_j)} \\
&= \frac{p_m^{x_m}(1 - p_m)^{1 - x_m} P(S_{m+1} = s - \sum_{j=1}^{m} x_j)}{P(S_m = s - \sum_{j=1}^{m-1} x_j)},
\end{aligned}
$$

The distributions of the partial sums $S_1, \ldots, S_n$ can be calculated before running the simulations using *generating functions*. See Exercise 2.10.

Conditioning on the sum (cont.)

NOTE: We only calculate the conditional probabilities we actually need along the way during the simulations, not the entire set of all possible conditional probabilities corresponding to all possible combinations of values of the X_j's.

Thus, in each simulation we calculate n probabilities, one for each X_j.

Each probability is calculated using a fixed number of operations (independent of n).

Hence, sampling from the conditional distribution of $\boldsymbol{X}$ given $S = s$, can be done in $O(n)$ time, i.e., just as fast as sampling from the unconditional distribution.

Conditioning on the sum (cont.)

When choosing the sample sizes $N_0, \ldots, N_n$, we may let $N_s \approx N \cdot P(S = s)$, $s = 0, 1, \ldots, n$.

However, it is possible to improve the results slightly. By examining the system, it is often easy to determine the size of the smallest minimal path and cut sets, and we introduce:

$$d = \text{The size of the smallest minimal path set}$$

$$c = \text{The size of the smallest minimal cut set}$$

We then have that $\theta_s = 0$ for $s < d$, and $\theta_s = 1$ for $s > n - c$, implying that $\mathrm{Var}(\phi | S = s) = 0$ for $s < d$, or $s > n - c$.

Hence, there is no point in spending simulations on estimating θ_s for $s < d$, or $s > n - c$, so we let $N_s = 0$ for $s < d$, or $s > n - c$. As a result, we have more simulations to spend on the remaining quantities.

Conditioning on the sum (cont.)

An extreme situation occurs when the system is a k-out-of-n-system, i.e., when $\phi(\boldsymbol{X}) = I(S \geq k)$. For such systems $d = k$, and $c = (n - k + 1)$.

This implies that $n - c = n - (n - k + 1) = k - 1$.

Hence, $\theta_s = 0$ for $s < k$ and $\theta_s = 1$ for $s \geq k$.

Thus, the CMC-estimate is equal to the true value of the reliability, and can be calculated without doing *any simulations at all*.

Conditioning on the sum (cont.)

For all other nontrivial systems, however, it is easy to see that we always have that $d \leq n - c$. In order to ensure that we get improved results, we assume that $P(d \leq S \leq n - c) > 0$, and let:

$$N_s \approx N \cdot P(S = s)/P(d \leq S \leq n - c), \quad s = d, \ldots, n - c\,.$$

We can now show that:

$$\mathrm{Var}(\hat{h}_{CMC}) \leq P(d \leq S \leq n - c)\,\mathrm{Var}(\hat{h}_{MC})$$

Hence, if d and $n - c$ are close, the variance is reduced considerably.

Section 7.3

System reliability when all the component state variables have identical reliabilities

Identical component reliabilities

If $p_1 = \cdots = p_n = p$, it is possible to improve things even further. Then S has a binomial distribution, and the conditional distribution of $\boldsymbol{X}$ given S is given by:

$$P(\boldsymbol{X} = \boldsymbol{x} \mid S = s) = \frac{p^{\sum_{i=1}^n x_i}(1-p)^{n-\sum_{i=1}^n x_i} \cdot I(\sum_{i=1}^n x_i = s)}{\binom{n}{s} p^s (1-p)^{n-s}} = \frac{1}{\binom{n}{s}} ,$$

for all $\boldsymbol{x}$ such that $\sum_{i=1}^n x_i = s$.

From this it follows that:

$$\theta_s = E[\phi \mid S = s] = \sum_{\{\boldsymbol{X} \mid \sum_{i=1}^n x_i = s\}} \phi(\boldsymbol{x}) P(\boldsymbol{X} = \boldsymbol{x} \mid S = s) = \frac{b_s}{\binom{n}{s}} ,$$

where b_s is the number of path sets with s components, $s = 0, \ldots, n$.

Identical component reliabilities (cont.)

The system reliability, h, expressed as a function of p, is given by:

$$h(p) = \sum_{s=0}^{n} \theta_s P(S = s) = \sum_{s=0}^{n} \frac{b_s}{\binom{n}{s}} \binom{n}{s} p^s (1-p)^{n-s}.$$

NOTE 1: $\theta_0, \ldots, \theta_n$ do not depend on p. Thus, by estimating these quantities, we get an estimate of $h(p)$ for all $p \in [0, 1]$.

NOTE 2: θ_s can be interpreted as the fraction of path sets of size s among all sets of size s. Thus, θ_s can be estimated by sampling random sets of size s and calculating the frequency of path sets among the sampled sets.

NOTE 3: Assuming that (C, ϕ) is non-trivial we have $\theta_0 = 0$. Thus, we can focus on estimating $\theta_1, \ldots, \theta_n$.

Identical component reliabilities (cont.)

In the following we let:

$$\phi(A) = \phi(\mathbf{1}^A, \mathbf{0}), \quad \text{for all } A \subseteq C.$$

ALGORITHM: For $i = 1, \ldots, N$ do:

STEP 1. Generate a random permutation $P_i = (c_{i,1}, \ldots, c_{i,n})$ of the component set C.

STEP 2. Let $A_{i,s} = \{c_{i,1}, \ldots, c_{i,s}\}, \quad s = 1, \ldots, n.$

STEP 3. Calculate $\phi(A_{i,s}), \quad s = 1, \ldots, n.$

Note that for $s = 1, \ldots, n$, $A_{i,s}$ is a random set of size s, $i = 1, \ldots, N$. Hence, unbiased estimators for $\theta_1, \ldots, \theta_n$ are given by:

$$\hat{\theta}_s = \frac{1}{N} \sum_{i=1}^{N} \phi(A_{i,s}), \quad s = 1, \ldots, n.$$

Identical component reliabilities (cont.)

Using the estimates $\hat{\theta}_1, \ldots, \hat{\theta}_n$ as well as $\hat{\theta}_0 = 0$, we get the following improved conditional Monte Carlo estimate:

$$\hat{h}_{CMC}(p) = \sum_{s=0}^{n} \hat{\theta}_s P(S = s)$$

$$= \sum_{s=0}^{n} \hat{\theta}_s \binom{n}{s} p^s (1-p)^{n-s}$$

NOTE: With this method all the θ_s's are estimated from the full sample of N. Thus, we do not need to partition the sample into subsamples.

The downside is that $\hat{\theta}_1, \ldots, \hat{\theta}_n$ become dependent. This makes it more difficult to calculate the resulting variance of $\hat{h}_{CMC}(p)$.

Identical component reliabilities (cont.)

As a comparison we describe how we can estimate $h(p)$ using a crude Monte Carlo method.

If $U \sim \text{Unif}(0,1)$, we have:

$$P(U \leq p) = p, \quad 0 \leq p \leq 1.$$

Hence, if $X(p) = I(U \leq p)$, it follows that $X(p) \sim \text{Binary}(p)$, and in particular:

$$P(X(p) = 1) = P(U \leq p) = p.$$

To generate $X_1(p), \ldots, X_N(p)$ so that $X_i(p) \sim \text{Binary}(p)$, $i = 1, \ldots, N$, we generate $U_1, \ldots, U_N$, where $U_i \sim \text{Unif}(0,1)$, $i = 1, \ldots, N$, and then let:

$$X_i(p) = I(U_i \leq p), \quad i = 1, \ldots, N.$$

Identical component reliabilities (cont.)

Now let (C, ϕ) be a binary monotone system where $C = \{1, \ldots, n\}$, and let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be the vector of the component state variables. We also assume that:

$$P(X_j = 1) = p, \quad j = 1, \ldots, n.$$

In order to estimate $h(p) = E[\phi(\boldsymbol{X})]$ as a function of $p \in [0, 1]$, we generate N vectors of independent uniformly distributed variables:

$$\boldsymbol{U}_i = (U_{i,1}, \ldots, U_{i,n}), \quad i = 1, \ldots, N,$$

and let $\boldsymbol{X}_i(p) = (X_{i,1}(p), \ldots, X_{i,n}(p))$, $i = 1, \ldots, N$ by:

$$X_{i,j}(p) = I(U_{i,j} \leq p), \quad i = 1, \ldots, N, \ j = 1, \ldots, n.$$

Identical component reliabilities (cont.)

We then estimate $h(p) = E[\phi(\boldsymbol{X})]$ as a function of $p \in [0, 1]$ by:

$$\hat{h}_{MC}(p) = \frac{1}{N} \sum_{i=1}^{N} \phi(\boldsymbol{X}_i(p))$$

Since $X_{i,j}(p)$ is non-decreasing in p and ϕ is non-decreasing in each argument, it follows that $\hat{h}_{MC}(p)$ is a non-decreasing function of p.

When running the simulation, it is not possible to store $\phi(\boldsymbol{X}_i(p))$ for all values of $p \in [0, 1]$. Instead, we let:

$$0 = p_0 \leq p_1 \leq \cdots \leq p_K = 1$$

where K is sufficiently large, and estimate:

$$\hat{h}_{MC}(p_r) = \frac{1}{N} \sum_{i=1}^{N} \phi(\boldsymbol{X}_i(p_r)), \quad r = 0, 1, \ldots, K.$$