My Personal View on Monte Carlo Simulation:  

Plenary Talk Presented at ASOR, Melbourne, 2007

Reuven Rubinstein

Faculty of Industrial Engineering and Management,  
Technion, Israel
Contents

1. The Breakthrough in the Field.
2. Importance Sampling (IS) and Cross-Entropy (CE).
3. The Inverse Transform Likelihood Ratio (ITLR) Method.
5. Stochastic Counterpart Versus Stochastic Approximation.
6. Rare-Event Simulation via the Cross-Entropy Method and MinxEnt.
7. Solving NP-hard Combinatorial Optimization Problems via Rare-Event Simulation.
8. Counting Problems via Rare-Events.
9. Conclusions and Open Problems.
Matters

Matters


**Special Issue:** *Annals of Operations Research*, 2005

My Personal View on Monte Carlo Simulation: *Plenary Talk Presented at ASOR, Melbourne, 2007*
Matters


Special Issue: *Annals of Operations Research*, 2005

The CE home page: http://www.cemethod.org
The Break Through in the Field

NO SLAIDS ON THIS TOPIC
My intention is to give a *simple* introduction to my personal view on Monte Carlo methods. But as Einstein said:

**Every thing should be made as simple as possible but not simpler.**
Let the expected performance of a stochastic system be

\[ \ell = \mathbb{E}_f[H(X)] = \int H(x) f(x) \, dx. \]

Here \( H \) is the sample performance function, \( f \) is the density of \( X \) and the subscript \( f \) in \( \mathbb{E}_f[H(X)] \) means that the expectation is taken with respect to the density \( f \).

An example of \( H(X) \) is an indicator function

\[ H(X) = I_{\{S(X) \geq \gamma\}} = \begin{cases} 1 & \text{if } S(X) \geq \gamma \\ 0 & \text{otherwise} \end{cases}. \]
We can write $\ell = \mathbb{E}_f[H(X)]$ as

$$\ell = \int H(x) \frac{f(x)}{g(x)} g(x) \, dx = \mathbb{E}_g \left[ H(X) \frac{f(X)}{g(X)} \right],$$

where the subscript $g$ means that the expectation is taken with respect to the IS pdf $g$.

The (unbiased) likelihood ratio estimator of $\ell$ is

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^{N} H(X_i) W(X_i),$$

where $W(x) = f(x)/g(x)$ is the likelihood ratio (LR), $X_1, \ldots, X_N \sim g$ and $g(x)$ is called importance sampling (IS) density.
Example: Counting via Monte Carlo

We start with the following basic Example.
Assume we want to calculate an area of some “irregular” region $\mathcal{X}^*$. The Monte-Carlo method suggests inserting the ”irregular" region $\mathcal{X}^*$ into a nice “regular" one $\mathcal{X}$ as per figure below.

$\mathcal{X}^*$ : Set of objects (paths in a graph, colorings of a graph, etc.)
$\mathcal{X}^*$ : Subset of special objects (cycles in a graph, colorings of a certain type, etc.).
Counting via Monte Carlo

To calculate $|\mathcal{X}^*|$ we apply the following sampling procedure:

(i) Generate a random sample $X_1, \ldots, X_N$, uniformly distributed over the “regular” region $\mathcal{X}$.

(ii) Estimate the desired area $|\mathcal{X}^*|$ as

$$|\mathcal{X}^*| = \hat{\ell}|\mathcal{X}|,$$

where

$$\hat{\ell} = \frac{N_{\mathcal{X}^*}}{N_{\mathcal{X}}} = \frac{1}{N} \sum_{k=1}^{N} I\{x_k \in \mathcal{X}^*\},$$

$I\{x_k \in \mathcal{X}^*\}$ denotes the indicator of the event $\{X_k \in \mathcal{X}^*\}$ and

$\{X_k\}$ is a sample from $f(x)$ over $\mathcal{X}$, where $f(x) = \frac{1}{|\mathcal{X}|}$.
The above formula

\[ |\hat{\mathcal{X}}^*| = \ell|\mathcal{X}| \]

is also valid for counting problems, that is where \( \mathcal{X}^* \) presents a discrete rather a continuous set of points. For example, in HC problem

1. \( \mathcal{X} \) is the entire set of tours in the graph. Note that
   \[ |\mathcal{X}| = (n - 1)! \]
2. \( \mathcal{X}^* \) is the subset of tours of length \( n \).
Counting via Rare-Events

Note that for counting problems

\[ \ell = \frac{|\mathcal{X}^*|}{|\mathcal{X}|} = \mathbb{E}_u[I\{x \in \mathcal{X}^*\}] \]

is typically very small, so the naive, crude Monte Carlo estimator of \( \ell \) is useless. It is easy to show that using importance sampling we obtain

\[ |\mathcal{X}^*| = \mathbb{E}_g[I\{x \in \mathcal{X}^*\} \frac{1}{g(X)}]. \]

The IS estimate of \( |\mathcal{X}^*| \) is therefore

\[ \hat{|\mathcal{X}^*|} = \frac{1}{N} \sum_{k=1}^{N} I\{x_k \in \mathcal{X}^*\} \frac{1}{g(X_k)} = \sum_{x_k \in \mathcal{X}^*} \frac{1}{g(X_k)}. \]
The best choice for \( g \) is, clearly, \( g^*(x) = 1/|\mathcal{X}^*|, \ x \in \mathcal{X}^* \), which is the \textit{uniform distribution on} \( \mathcal{X}^* \). Under \( g^* \) the estimator has \textbf{zero variance}, since the random variable \( \hat{|\mathcal{X}^*|} = \text{const} \), so that only one sample is required. However, sampling from such \( g^* \) is impractical, since it requires availability of our target value \( |\mathcal{X}^*| \).

To overcome this difficulty we shall show in the following sections how to construct "good" (low variance) IS sample pdf’s \( g(x) \) (parametric and nonparametric) for different \#P-complete counting problems.
We shall consider now only the case where the IS density $g$ belongs to the same parametric family as $f$. Let $f(\cdot; u)$ denote the density of the random vector $X$ for some fixed “nominal” parameter $u \in V$.

In this case the LR estimator $\hat{\ell}$, with $g(x) = f(x; v)$ becomes

$$
\hat{\ell} = \frac{1}{N} \sum_{i=1}^{N} H(X_i) W(X_i; u, v), \quad \text{and} \quad W(X; u, v) = \frac{f(X; u)}{f(X; v)},
$$

where $X_1, \ldots, X_N$ is a random sample from $f(\cdot; v)$. 

My Personal View on Monte Carlo Simulation: *Plenary Talk Presented at ASOR, Melbourne, 2007* – p.14
The “optimal” $\nu = \ast \nu$ should minimize

$$\text{Var}_\nu [H(X) W(X; u, \nu)],$$

which is the same as minimizing

$$V(\nu) = \mathbb{E}_u [H^2(X) W(X; u, \nu)].$$

This function may still be difficult to minimize.
The stochastic counterpart of

$$\min_{\nu} V(\nu) = \min_{\nu} \mathbb{E}_u \left[ H^2(X) W(X; u, \nu) \right].$$

is

$$\min_{\nu} \hat{V}(\nu) = \min_{\nu} \frac{1}{N} \sum_{i=1}^{N} H^2(X_i) W(X_i; u, \nu),$$

where $X_1, \ldots, X_N$ is an i.i.d. sample from $f(\cdot; u)$.

Hence, we can estimate $\nu_*$ by minimizing $\hat{V}(\nu)$. This usually involves numerical minimization.
In the Cross-Entropy method we choose \( g = f(\cdot; \nu) \) such that the “distance” between the densities \( g^* \) and \( f(\cdot; \nu) \) is minimal.

The **Kullback-Leibler** or cross-entropy distance is defined as:

\[
D(g, h) = \mathbb{E}_g \left[ \log \frac{g(X)}{h(X)} \right] = \int g(x) \log g(x) \, dx - \int g(x) \log h(x) \, dx.
\]

Note that \( D(g, h) \geq 0 \) and \( D(g, h) = 0 \) when \( g = h \).

**Shannon Entropy**

\[
\mathcal{H}(x) = -\mathbb{E} \log f(X) = -\int f(x) \log f(x) \, dx
\]
The Parametric CE Method

This is equivalent to solving

$$\max_v D(v) = \max_v \mathbb{E}_u [H(X) \log f(X; v)] .$$

We may \textit{estimate} the optimal solution $v^*$ by solving the following stochastic counterpart:

$$\max_v \frac{1}{N} \sum_{i=1}^{N} I\{S(X_i) \geq \gamma\} \log f(X_i; v) ,$$

where $X_1, \ldots, X_N$ is a random sample from $f(\cdot; u)$. 
Summarizing: in CE we maximize

\[
\max_v \hat{D}(v) = \max_v \frac{1}{N} \sum_{i=1}^{N} H(X_i) \ln f(X_i; v),
\]

instead of

\[
\min_v \hat{V}(v) = \min_v \frac{1}{N} \sum_{i=1}^{N} H^2(X_i) W(X_i; u, v),
\]

where \( X_i \sim f(x; w) \).

The advantage: CE programs can be typically solved analytically, while the variance minimization only numerically.
The Inverse Transform Method

Consider estimation of \( \ell = \mathbb{E}_f[H(X)] \), where \( X \sim f(x) \).
According to the inverse transform (IT) method, we can write \( X \) (for the one-dimensional case) as

\[
X = F^{-1}(U),
\]

where \( U \sim U(0,1) \) and \( F^{-1} \) is the inverse of the cdf \( F \).

Substituting \( X = F^{-1}(U) \) into \( \ell = \mathbb{E}[H(X)] \) we obtain

\[
\ell = \mathbb{E}_U[H(F^{-1}(U))] = \mathbb{E}_U[\tilde{H}(U)].
\]

Note that the expectation in \( \ell = \mathbb{E}_f[H(X)] \) is taken with respect to \( f(x) \), while expectation in \( \ell = \mathbb{E}_U[\tilde{H}(U)] \) is taken with respect to the uniform \( U(0,1) \) distribution.
To estimate \( \ell = \mathbb{E}_U[\tilde{H}(U)] \) one can use IS. As IS pdf one can take on \((0, 1)\) any pdf \( h(u; \nu) \) parameterized by some reference parameter \( \nu \). An example is the \( Beta(\nu, 1) \)-distribution, with density
\[
h(u; \nu) = \nu u^{\nu-1}, \quad u \in (0, 1),
\]
with \( \nu > 0 \), or the \( Beta(1, \nu) \)-distribution, with density
\[
h(u; \nu) = \nu (1 - u)^{\nu-1}, \quad u \in (0, 1).
\]
ITLR Method

Using $Beta(1, \nu)$ as the IS pdf we can write $\ell$ as

$$\ell = \mathbb{E}_\nu[\tilde{H}(U) \tilde{W}(U; \nu)] ,$$

where $U \sim h(u; \nu)$ and $\tilde{W}(U; \nu) = \frac{1}{h(U; \nu)}$ is the likelihood ratio (LR). The LR estimator of $\ell$ is given by

$$\hat{\ell} = N^{-1} \sum_{i=1}^{N} \tilde{H}(U_i) \tilde{W}(U_i; \nu),$$

where $U_1, \ldots, U_N$ is a random sample from $h(u; \nu)$. We call $\hat{\ell}$ the inverse transform - likelihood ratio (ITLR) estimator. It is generic.
Suppose, for example, $X \sim \text{Weibull}(\alpha, \lambda)$, that is, $X$ has the density

$$f(x; \alpha, \lambda) = \alpha \lambda (\lambda x)^{\alpha-1} e^{-(\lambda x)^\alpha}.$$ 

Note that a Weibull random variable can be generated using the transformation

$$X = \lambda^{-1} Z^{1/\alpha}$$

where $Z$ is a random variable distributed $\text{Exp}(1)$. Applying the IT method we obtain $X = F^{-1}(U) = \lambda^{-1}(-\ln(1-U))^{1/\alpha}$, and, thus

$$\hat{\ell} = N^{-1} \sum_{i=1}^{N} H(\lambda^{-1} (-\ln(1 - U_i))^{1/\alpha})/h(U_i; \nu).$$
We consider the case where the expected performance is given by

\[ \ell(u) = \mathbb{E}_u[H(X)] = \int H(x)f(x, u)dx, \]

with \( X \sim f(x, u) \) and the sensitivity is performed with respect to \( u \). We shall introduce the celebrated score function (SF) method for sensitivity analysis of discrete-event static system. The goal of the SF method is to estimate the gradient and higher derivatives of \( \ell(u) \) with respect to the distributional parameter vector \( u \).
Consider first the case where $u$ is scalar. Then under mild conditions the differentiation and expectation (integration) operators are interchangeable we have that

$$\frac{d\ell(u)}{du} = \frac{d}{du} \int H(\mathbf{x}) f(\mathbf{x}; u) d\mathbf{x} = \int H(\mathbf{x}) \frac{df(\mathbf{x}; u)}{du} d\mathbf{x}$$

$$= \int H(\mathbf{x}) \frac{df(\mathbf{x}; u)}{du} \frac{d \log f(\mathbf{x}; u)}{du} f(\mathbf{x}; u) d\mathbf{x} = \mathbb{E}_u \left[ H(\mathbf{X}) \frac{d \log f(\mathbf{X}; u)}{du} \right]$$

$$= \mathbb{E}_u \left[ H(\mathbf{X}) S(u; \mathbf{X}) \right],$$

where

$$S(u; \mathbf{x}) = \frac{d \log f(\mathbf{x}; u)}{du}$$

is called the score function \text{(SF)}. 

\text{My Personal View on Monte Carlo Simulation: Plenary Talk Presented at ASOR, Melbourne, 2007 – p.25}
Consider next the multidimensional case. We have for the gradient and the higher order derivatives of $\ell(u)$

$$\nabla^k \ell(u) = \mathbb{E}_u \left[ H(X) S^{(k)}(u; X) \right],$$

where

$$S^{(k)}(u; x) = \frac{\nabla^k f(x; u)}{f(x; u)}$$

is the $k$-th order score function $k$-th order $k = 0, 1, 2, \ldots$. In particular, $S^{(0)}(u; x) = 1$ (by definition) and $S^{(1)}(u; x) = S(u; x) = \nabla \log f(x; u)$. 
In general, the quantities $\nabla^k \ell(u)$, $k = 0, 1, \ldots$, are not available analytically, since the response $\ell(u)$ is so. They can be estimated, however, via simulation as

$$\hat{\nabla}^k \ell(u) = \frac{1}{N} \sum_{i=1}^{N} H(X_i) S^{(k)}(u; X_i).$$

It is readily seen that the function $\ell(u)$, and all the sensitivities $\nabla^k \ell(u)$ can be estimated from a single simulation, since all of them are expressed as expectations with respect to the same pdf, $f(x; u)$. 
Let $H(X) = X$, with $X \sim \text{Exp}(\lambda = u)$. This is a toy example since $\nabla \ell(u) = -1/u^2$. We have that $S(u; x) = u^{-1} - x$, and therefore

$$\hat{\nabla} \ell(u) = \frac{1}{N} \sum_{i=1}^{N} X_i (u^{-1} - X_i) \approx -\frac{1}{u^2}$$

is an estimator of $\nabla \ell(u)$, where $X_1, \ldots, X_N$ is a random sample from $\text{Exp}(u)$. 
Applying importance sampling (IS) \( g(x) \) to
\[ \nabla^k \ell(u) = \mathbb{E}_u[H(X)S^{(k)}(u; X)] \] we obtain

\[ \nabla^k \ell(u) = \mathbb{E}_g[H(X)S^{(k)}(u; X)] \frac{f(x; u)}{g(x)} \] (1)

The LR estimator of \( \nabla^k \ell(u) \) can be written as

\[ \hat{\nabla}^k \ell(u) = \frac{1}{N} \sum_{i=1}^{N} H(X_i)S^{(k)}(u; X_i) \frac{f(X_i; u)}{g(X_i)} \] , (2)

where \( X_1, \ldots, X_N \) is a random sample from \( g(x) \).
This means that by varying \( u \) and keeping \( g \) fixed we can, in principle, estimate unbiasedly the whole \textit{response surface} \( \{\nabla^k \ell(u), u \in V\} \) from a \textit{single simulation}. Often the IS distribution is chosen in the same class of distributions as the original one. That is, \( g(x) = f(x; v) \), for some \( v \in V \).
Consider
\[
\max_u \{ \ell(u) = \mathbb{E}_u [H(X)] = \int H(x) f(x; u) dx \}.
\]

We solve
\[
\nabla \ell(u) = 0 \rightarrow \int H(x) \nabla \log f(x; u) dx \mathbb{E}_u [H(X)] = 0.
\]

For example, considering the CE program
\[
\max_v D(v) = \max_v \mathbb{E}_u [H(X) \log f(X; u)] .
\]

we obtain
\[
\nabla D(v) = \mathbb{E}_u [H(X) \nabla \log f(X; u)] = 0.
\]
Monte Carlo Optimization

While solving the stochastic counterpart:

$$\max_v \left\{ \frac{1}{N} \sum_{i=1}^{N} I_{\{S(X_i) \geq \gamma\}} \log f(X_i; v) \right\},$$

where $X_1, \ldots, X_N$ is a random sample from $f(\cdot; u)$ we would simply solve

$$\frac{1}{N} \sum_{i=1}^{N} I_{\{S(X_i) \geq \gamma\}} \nabla \log f(X_i; v) = 0$$
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
- Simulation-based (noisy) Optimization, like Optimal Buffer Allocation and Optimization in Finance Engineering
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
- Simulation-based (noisy) Optimization, like Optimal Buffer Allocation and Optimization in Finance Engineering
- Multi-extremal Continuous Optimization
The Cross-Entropy Method: Applications

- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
- Simulation-based (noisy) Optimization, like Optimal Buffer Allocation and Optimization in Finance Engineering
- Multi-extremal Continuous Optimization
- NP-hard Counting problems: Hamiltonian Cycles, SAW’s, calculation the Permanent, Satisfiability Problem, etc.
We wish to color the nodes white and black.

How should we color so that the total number of links between the two groups is maximized? This problem is known as the Maximal Cut problem.
Continuous Optimization: A Multi-extremal function

This is the trigonometric function.

What is its global maximum, and where is it attained?
Another Multi-extremal function
Counting Hamiltonian Cycles

How many Hamiltonian cycles does this graph have?
Calculating the Number of HC’s

My Personal View on Monte Carlo Simulation: *Plenary Talk Presented at ASOR, Melbourne, 2007* – p.39
Hamiltonian Cycles: Large Example.

Performance of CE Algorithm for \( n = 100, \eta = 0.4 \) and \( N = 5n^2 = 50,000 \)

| \( t \) | \( |X^*| \) Average | Min | Max | \( \bar{\varepsilon} \) | \( \varepsilon_* \) | \( \varepsilon^* \) | \( RE \) |
|---|---|---|---|---|---|---|---|
| 0  | 7.42E+115 | 6.14E+115 | 8.21E+115 | 0.056 | 0.005 | 0.173 | 0.077 |
| 1  | 8.00E+115 | 6.79E+115 | 8.97E+115 | 0.074 | 0.001 | 0.151 | 0.089 |
| 2  | 7.12E+115 | 6.55E+115 | 7.95E+115 | 0.053 | 0.007 | 0.116 | 0.068 |
| 3  | 7.48E+115 | 6.79E+115 | 8.12E+115 | 0.048 | 0.003 | 0.092 | 0.059 |
| 4  | 7.84E+115 | 6.76E+115 | 8.70E+115 | 0.058 | 0.005 | 0.137 | 0.078 |

My Personal View on Monte Carlo Simulation: *Plenary Talk Presented at ASOR, Melbourne, 2007* – p.40
The CE Method can be used to solve two types of problems:

1. **Estimation:**

   \[
   \text{Estimate } \ell = \mathbb{E}[H(X)]
   \]

   \(X\): random vector/process taking values in some set \(\mathcal{X}\).

   \(H\): function on \(\mathcal{X}\).

   In particular, the estimation of *rare event* probabilities:

   \[
   \ell = \mathbb{P}(S(X) \geq \gamma), \text{ where } S \text{ is another function on } \mathcal{X}.
   \]
Introduction

The CE Method can be used to solve two types of problems:

1. **Estimation**:

   Estimate \( \ell = \mathbb{E}[H(X)] \)

   \( X \): random vector/process taking values in some set \( \mathcal{X} \).

   \( H \): function on \( \mathcal{X} \).

   In particular, the estimation of *rare event* probabilities:

   \( \ell = \mathbb{P}(S(X) \geq \gamma) \), where \( S \) is another function on \( \mathcal{X} \).

2. **Optimization**:

   Determine \( \max_{x \in \mathcal{X}} S(x) \)
The Simplest Example

Consider the problem of estimating

$$\ell = \mathbb{E}_u \left[ I_{\{X \geq \gamma\}} \right] = \mathbb{P}(X \geq \gamma),$$

with $X \sim \text{Exp}(1/u)$. Consider the family of IS densities

$$f(x; v) = v^{-1} e^{-x/v}, \quad x \geq 0.$$

To find the CE optimal parameter $v^*$ we need to maximize $D(v)$, with

$$D(v) = \mathbb{E}_u \left[ I_{\{X \geq \gamma\}} W(X; u, w) \log f(X; v) \right]$$

$$= \mathbb{E}_u \left[ I_{\{X \geq \gamma\}} W(X; u, w) \left( -\log(v) - \frac{X}{v} \right) \right].$$
Setting \( D'(v) = 0 \) we find

\[
v^* = \frac{\mathbb{E}_u \left[ I_{\{X \geq \gamma\}} X \right]}{\mathbb{E}_u \left[ I_{\{X \geq \gamma\}} \right]} = \frac{\mathbb{E}_w \left[ I_{\{X \geq \gamma\}} W(X; u, w) X \right]}{\mathbb{E}_w \left[ I_{\{X \geq \gamma\}} W(X; u, w) \right]}.\]

Similarly, the solution to \( \hat{D}'(v) = 0 \) is

\[
\hat{v} = \frac{\sum_{i=1}^{N} I_{\{X_i \geq \gamma\}} W(X_i; u, w) X_i}{\sum_{i=1}^{N} I_{\{X_i \geq \gamma\}} W(X_i; u, w)},
\]

with \( X_1, \ldots, X_N \sim f(\cdot; w) \).
Generate $\gamma_1, v_1, \gamma_2, v_2, \ldots, \gamma_T, v_T$.

- $t = 0$: $f(x; u) = u^{-1} e^{-u^{-1}x}$
- $t = 1$: $f(x; v_1) = v_1^{-1} e^{-v_1^{-1}x}$
- $t = 2$: $f(x; v_2) = v_2^{-1} e^{-v_2^{-1}x}$
- $t = 3$: $f(x; w) = w^{-1} e^{-w^{-1}x}$
Generic CE Algorithm

The CE Algorithm generates a sequence \( \{(\hat{\gamma}_t, \hat{v}_t)\}, \ t \geq 0; \)

- \( \hat{\gamma}_t \) monotonically increases and crosses the level \( \gamma \) after a finite number of iterations \( t \),
- \( \hat{v}_t \) converges to the optimal parameter \( v^* \) of the IS density \( g(x) = f(x; v^*) \).

Assuming independent components from a 1-parameter exponential family parameterized by the mean, the analytic updating formula is

\[
\hat{v}_{t,j} = \frac{\sum_{i=1}^{N} I\{S(X_i) \geq \hat{\gamma}_t\} W(X_i; u, \hat{v}_{t-1}) X_{ij}}{\sum_{i=1}^{N} I\{S(X_i) \geq \hat{\gamma}_t\} W(X_i; u, \hat{v}_{t-1})}.
\]
Let $\mathcal{X}$ be a finite set of states, and let $S$ be a real-valued sample function $S$ over $\mathcal{X}$. We wish to find

$$S(x^*) = \gamma^* = \max_{x \in \mathcal{X}} S(x).$$

The starting point in the methodology of the CE method is to associate with the above optimization problem a meaningful estimation problem.
General CE Procedure

The CE method cast the original optimization problem of $S(x)$ into an associated rare-events probability estimation problem, that estimation of

$$\ell = \mathbb{P}(S(X) \geq \gamma) = \mathbb{E}[I_{\{S(x) \geq \gamma\}}].$$

and involves the following iterative steps:

- Formulate a parameterized random mechanism to generate the objects $x \in \mathcal{X}$. 

My Personal View on Monte Carlo Simulation: Plenary Talk Presented at ASOR, Melbourne, 2007 – p.47
General CE Procedure

The CE method casts the original optimization problem of $S(x)$ into an associated rare-events probability estimation problem, that estimation of

$$\ell = \mathbb{P}(S(X) \geq \gamma) = \mathbb{E}[I_{\{S(x) \geq \gamma\}}].$$

and involves the following iterative steps:

- Formulate a parameterized random mechanism to *generate* the objects $x \in \mathcal{X}$.

- Give the *updating formulas* for the parameters of the random mechanism (obtained via Cross-Entropy minimization), in order to produce a better sample in the next iteration.
Max-Cut Example

Consider a weighted graph $G$ with node set $V = \{1, \ldots, n\}$. Partition the nodes of the graph into two subsets $V_1$ and $V_2$ such that the sum of the weights of the edges going from one subset to the other is maximized.

Example

![Graph Diagram]
The Max-Cut problem

Cost matrix:

\[
C = \begin{pmatrix}
0 & c_{12} & c_{13} & 0 & 0 & 0 \\
0 & c_{21} & 0 & c_{23} & c_{24} & 0 \\
c_{31} & c_{32} & 0 & c_{34} & c_{35} & 0 \\
0 & c_{42} & c_{43} & 0 & c_{45} & c_{46} \\
0 & 0 & c_{53} & c_{54} & 0 & c_{56} \\
0 & 0 & 0 & c_{64} & c_{65} & 0
\end{pmatrix}.
\]

\(\{V_1, V_2\} = \{\{1, 3, 4\}, \{2, 5, 6\}\}\) is a possible cut. The cost of the cut is

\[c_{12} + c_{32} + c_{35} + c_{42} + c_{45} + c_{46}.\]
Generation of cut vectors: The most natural and easiest way to generate the cut vectors is to let $X_2, \ldots, X_n$ be independent Bernoulli random variables with success probabilities $p_2, \ldots, p_n$. 
Generation and Updating Formulas

**Generation of cut vectors:** The most natural and easiest way to generate the cut vectors is to let $X_2, \ldots, X_n$ be independent Bernoulli random variables with success probabilities $p_2, \ldots, p_n$.

**Updating formulas:**

$$
\hat{p}_{t,j} = \sum_{i=1}^{N} \frac{I\{S(x_i) \geq \hat{\gamma}_t\} I\{x_{ij} = 1\}}{\sum_{i=1}^{N} I\{S(x_i) \geq \hat{\gamma}_t\}} = \sum_{x_i \in \mathcal{E}_t} \frac{X_{ij}}{|\mathcal{E}_t|}, \quad j = 2, \ldots, n
$$

where $|\mathcal{E}_t| = \rho N$, the number of elite samples.

Note that the likelihood term is missing.
1 Start with $\hat{p}_0 = (1/2, \ldots, 1/2)$. Let $t := 1$. 
1 Start with \( \hat{p}_0 = (1/2, \ldots, 1/2) \). Let \( t := 1 \).

2 **Update** \( \hat{\gamma}_t \): Draw \( X_1, \ldots, X_N \) from \( \text{Ber}(\hat{p}_t) \). Let \( \hat{\gamma}_t \) be the worst performance of the \( \rho \times 100\% \) best performances.
1 Start with $\hat{p}_0 = (1/2, \ldots, 1/2)$. Let $t := 1$.

2 Update $\hat{\gamma}_t$: Draw $X_1, \ldots, X_N$ from $\text{Ber}(\hat{p}_t)$. Let $\hat{\gamma}_t$ be the worst performance of the $\rho \times 100\%$ best performances.

3 Update $\hat{p}_t$: Use the same sample to calculate

$$
\hat{p}_{t,j} = \frac{\sum_{X_i \in E_t} X_{ij}}{|E_t|},
$$

$j = 1, \ldots, n$, where $X_i = (X_{i1}, \ldots, X_{in})$, and increase $t$ by 1.
CE Optimization Algorithm

1 Start with $\hat{p}_0 = (1/2, \ldots, 1/2)$. Let $t := 1$.

2 Update $\hat{\gamma}_t$: Draw $X_1, \ldots, X_N$ from Ber($\hat{p}_t$). Let $\hat{\gamma}_t$ be the worst performance of the $\rho \times 100\%$ best performances.

3 Update $\hat{p}_t$: Use the same sample to calculate

$$\hat{p}_{t,j} = \frac{\sum_{X_i \in E_t} X_{ij}}{|E_t|},$$

$j = 1, \ldots, n$, where $X_i = (X_{i1}, \ldots, X_{in})$, and increase $t$ by 1.

4 If the stopping criterion is met, then stop; otherwise set $t := t + 1$ and reiterate from step 2.
Performance of the CE Algorithm for the knapsack problem with the instance matrix $A = (20 \times 11)$ and $N = 10,000$. This problem was taken from the website http://elib.zib.de. Using full enumeration we found that the total number of multiple extrema is 612.

<table>
<thead>
<tr>
<th>$t$</th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
<th>PV</th>
<th>RE</th>
<th>S</th>
<th>$m$</th>
<th>$RD_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>639.6</td>
<td>943.7</td>
<td>419.4</td>
<td>0.00</td>
<td>0.225</td>
<td>6.93</td>
<td>10</td>
<td>55.73</td>
</tr>
<tr>
<td>1</td>
<td>619.2</td>
<td>697.6</td>
<td>564.8</td>
<td>0.03</td>
<td>0.072</td>
<td>5.78</td>
<td>11</td>
<td>0.02</td>
</tr>
<tr>
<td>2</td>
<td>630.8</td>
<td>706.5</td>
<td>557.0</td>
<td>0.07</td>
<td>0.059</td>
<td>5.18</td>
<td>11</td>
<td>0.03</td>
</tr>
<tr>
<td>3</td>
<td>628.1</td>
<td>698.1</td>
<td>533.2</td>
<td>0.08</td>
<td>0.083</td>
<td>4.95</td>
<td>11</td>
<td>0.03</td>
</tr>
<tr>
<td>4</td>
<td>573.7</td>
<td>671.2</td>
<td>504.9</td>
<td>0.09</td>
<td>0.083</td>
<td>4.88</td>
<td>11</td>
<td>0.06</td>
</tr>
<tr>
<td>5</td>
<td>599.3</td>
<td>719.6</td>
<td>525.7</td>
<td>0.09</td>
<td>0.100</td>
<td>4.72</td>
<td>11</td>
<td>0.03</td>
</tr>
<tr>
<td>6</td>
<td>576.9</td>
<td>646.4</td>
<td>508.0</td>
<td>0.09</td>
<td>0.071</td>
<td>4.76</td>
<td>11</td>
<td>0.06</td>
</tr>
</tbody>
</table>
A typical dynamics of the CE Algorithm for the knapsack problem with the instance matrix $A = (20 \times 11)$ and $N = 10,000$. 

My Personal View on Monte Carlo Simulation: Plenary Talk Presented at ASOR, Melbourne, 2007 – p.53
The Screening Method

Consider estimation of

\[ \ell = \mathbb{P}_u(S(X) \geq \gamma) = \mathbb{E}_u \left[ I\{S(x) \geq \gamma\} \right] \]

for some fixed level \( \gamma \). As usual \( S(X) \) is the sample performance, \( X \) a random vector with pdf \( f(\cdot; u) \), belonging to some parametric family \( \{f(\cdot; v), v \in V\} \) and \( \{S(X) \geq \gamma\} \) is a rare event. We can estimate \( \ell \) using the LR estimator

\[ \hat{\ell} = \frac{1}{N} \sum_{k=1}^{N} I\{S(x_k) \geq \gamma\} W(X_k; u, v), \]

where \( X_1, \ldots, X_N \) is a random sample from \( f(x; v) \), and

\[ W(X_k; u, v) = f(X_k; u)/f(X_k; v) \]

is the likelihood ratio.
Example: Stochastic Shortest Path

Our objective is to efficiently estimate the probability $\ell$ that the shortest path from node $A$ to node $B$ in the network has a length of at least $\gamma$. The random lengths $X_1, \ldots, X_5$ of the links are assumed to be independent and exponentially distributed with means $u_1, \ldots, u_5$, respectively and

\[
S(X) = \min\{X_1 + X_4, X_1 + X_3 + X_5, X_2 + X_5, X_2 + X_3 + X_4\}.
\]
The Screening Method

The table below displays the performance of the CE algorithm.

Table 0: Convergence of the sequence \( \{(\hat{\gamma}_t, \hat{v}_t)\} \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( \hat{\gamma}_t )</th>
<th>( \hat{v}_t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>1.1656</td>
<td>1.9805</td>
</tr>
<tr>
<td>2</td>
<td>2.1545</td>
<td>2.8575</td>
</tr>
<tr>
<td>3</td>
<td>3.1116</td>
<td>3.7813</td>
</tr>
<tr>
<td>4</td>
<td>4.6290</td>
<td>5.2803</td>
</tr>
<tr>
<td>5</td>
<td>6.0000</td>
<td>6.7950</td>
</tr>
</tbody>
</table>
SUMMARY

Lemma 1: We have to learn to live with uncertainty

Theorem 1: We can model the uncertainty

Corollary 1: We can make a living out of uncertainty
THANK YOU