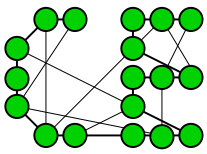


My Personal View on Monte Carlo Simulation:

Plenary Talk Presented at ASOR, Melbourne, 2007

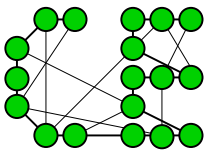
Reuven Rubinstein

Faculty of Industrial Engineering and Management,
Technion, Israel



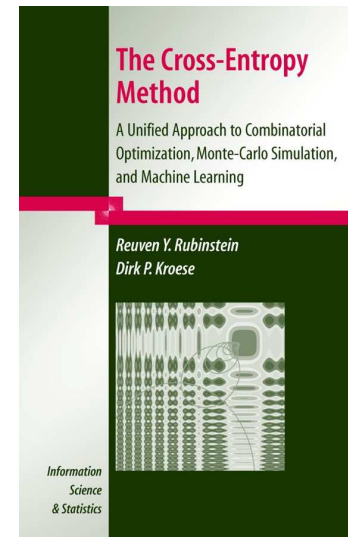
Contents

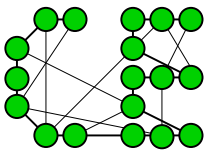
1. The Breakthrough in the Field.
2. Importance Sampling (IS) and Cross-Entropy (CE).
3. The Inverse Transform Likelihood Ratio (ITLR) Method.
4. Sensitivity Analysis Using the Score Function (SF) Method.
5. Stochastic Counterpart Versus Stochastic Approximation.
6. Rare-Event Simulation via the Cross-Entropy 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

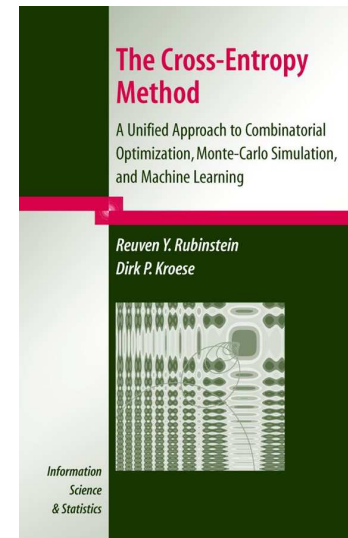
Book: R.Y. Rubinstein and D.P. Kroese.
The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte Carlo Simulation and Machine Learning, Springer-Verlag, New York, 2004.



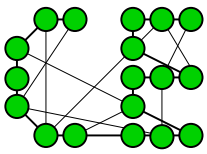


Matters

Book: R.Y. Rubinstein and D.P. Kroese.
The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte Carlo Simulation and Machine Learning, Springer-Verlag, New York, 2004.

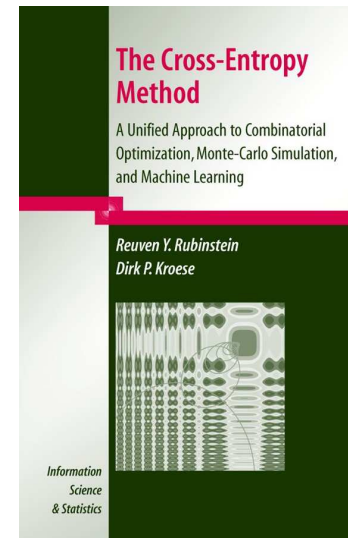


Special Issue: *Annals of Operations Research*, 2005



Matters

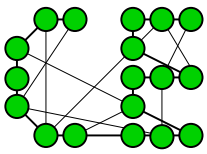
Book: R.Y. Rubinstein and D.P. Kroese.
The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte Carlo Simulation and Machine Learning, Springer-Verlag, New York, 2004.



Special Issue: *Annals of Operations Research*, 2005

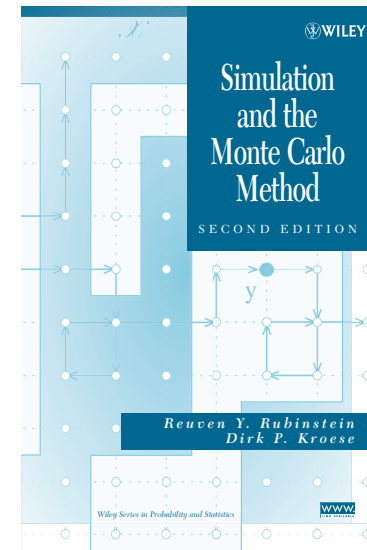
The CE home page:

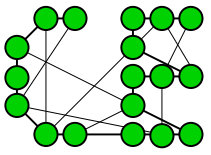
<http://www.cemethod.org>



CE Matters

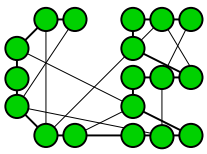
Book: R.Y. Rubinstein and D.P. Kroese.
*Simulation and the Monte Carlo Method:
Second Edition*, Wiley, 2007.





The Break Through in the Field

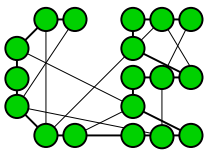
NO SLAIDS ON THIS TOPIC



Einstein's Statement

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.



Estimation via Importance Sampling

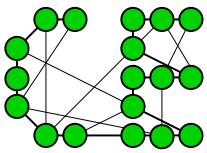
Let the expected performance of a stochastic system be

$$\ell = \mathbb{E}_f[H(\mathbf{X})] = \int H(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} .$$

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

An example of $H(\mathbf{X})$ is an **indicator function**

$$H(\mathbf{X}) = I_{\{S(\mathbf{x}) \geq \gamma\}} = \begin{cases} 1 & \text{if } S(\mathbf{X}) \geq \gamma \\ 0 & \text{otherwise .} \end{cases}$$



Importance sampling

We can write $\ell = \mathbb{E}_f[H(\mathbf{X})]$ as

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

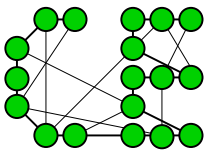
where the subscript g means that the expectation is taken with respect to the IS pdf g .

The (unbiased) **likelihood ratio estimator** of ℓ is

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

where $W(\mathbf{x}) = f(\mathbf{x})/g(\mathbf{x})$ is the likelihood ratio (LR),

$\mathbf{X}_1, \dots, \mathbf{X}_N \sim g$ and $g(\mathbf{x})$ is called **importance sampling** (IS) density.

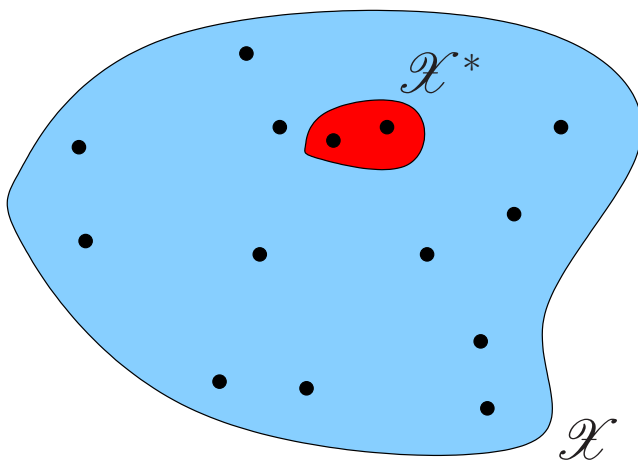


Example: Counting via Monte Carlo

We start with the following basic

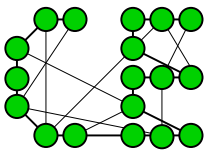
Example.

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:

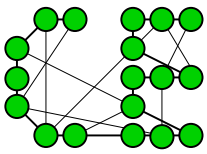
- (i) Generate a random sample $\mathbf{X}_1, \dots, \mathbf{X}_N$, *uniformly* distributed over the “regular” region \mathcal{X} .
- (ii) Estimate the desired area $|\mathcal{X}^*|$ as

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

where

$$\widehat{\ell} = \frac{N_{\mathcal{X}^*}}{N_{\mathcal{X}}} = \frac{1}{N} \sum_{k=1}^N I_{\{\mathbf{X}_k \in \mathcal{X}^*\}},$$

$I_{\{\mathbf{X}_k \in \mathcal{X}^*\}}$ denotes the indicator of the event $\{\mathbf{X}_k \in \mathcal{X}^*\}$ and $\{\mathbf{X}_k\}$ is a sample from $f(\mathbf{x})$ over \mathcal{X} , where $f(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$.



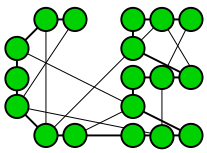
Counting via Monte Carlo

The above formula

$$|\widehat{\mathcal{X}^*}| = \widehat{\ell} |\mathcal{X}|$$

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

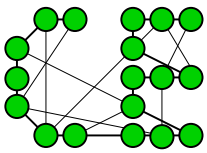
$$\ell = \frac{|\mathcal{X}^*|}{|\mathcal{X}|} = \mathbb{E}_{\mathbf{U}}[I_{\{\mathbf{X} \in \mathcal{X}^*\}}]$$

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

$$|\mathcal{X}^*| = \mathbb{E}_g \left[I_{\{\mathbf{X} \in \mathcal{X}^*\}} \frac{1}{g(\mathbf{X})} \right].$$

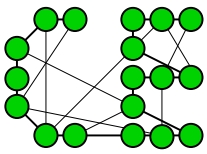
The IS estimate of $|\mathcal{X}^*|$ is therefore

$$|\widehat{\mathcal{X}^*}| = \frac{1}{N} \sum_{k=1}^N I_{\{\mathbf{X}_k \in \mathcal{X}^*\}} \frac{1}{g(\mathbf{X}_k)} = \sum_{\mathbf{X}_k \in \mathcal{X}^*} \frac{1}{g(\mathbf{X}_k)}.$$



Choosing the IS pdf $g(\boldsymbol{x})$

The best choice for g is, clearly, $g^*(\boldsymbol{x}) = 1/|\mathcal{X}^*|$, $\boldsymbol{x} \in \mathcal{X}^*$, which is the *uniform distribution on \mathcal{X}^** . Under g^* the estimator has **zero variance**, since the random variable $|\widehat{\mathcal{X}^*}| = \text{const}$, 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 $g(\boldsymbol{x})$ (parametric and nonparametric) for different #P-complete counting problems.



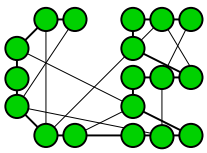
Parametric IS

We shall consider now only the case where the IS density g belongs to the **same parametric family** as f . Let $f(\cdot; \mathbf{u})$ denote the density of the random vector \mathbf{X} for some fixed “nominal” parameter $\mathbf{u} \in \mathcal{V}$.

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

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

where $\mathbf{X}_1, \dots, \mathbf{X}_N$ is a random sample from $f(\cdot; \mathbf{v})$.



Parametric IS

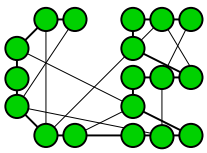
The “optimal” $\mathbf{v} = \ast \mathbf{v}$ should minimize

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

which is the same as minimizing

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

This function may still be difficult to minimize.



Parametric IS

The stochastic counterpart of

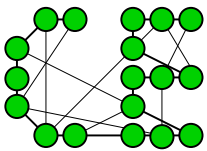
$$\min_{\mathbf{v}} V(\mathbf{v}) = \min_{\mathbf{v}} \mathbb{E}_{\mathbf{u}} [H^2(\mathbf{X}) W(\mathbf{X}; \mathbf{u}, \mathbf{v})] .$$

is

$$\min_{\mathbf{v}} \hat{V}(\mathbf{v}) = \min_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^N H^2(\mathbf{X}_i) W(\mathbf{X}_i; \mathbf{u}, \mathbf{v}),$$

where $\mathbf{X}_1, \dots, \mathbf{X}_N$ is an i.i.d. sample from $f(\cdot; \mathbf{u})$.

Hence, we can estimate $_*\mathbf{v}$ by minimizing $\hat{V}(\mathbf{v})$. This usually involves numerical minimization.



Cross-Entropy (CE) Method

In the Cross-Entropy method we choose $g = f(\cdot; \mathbf{v})$ such that the “distance” between the densities g^* and $f(\cdot; \mathbf{v})$ is minimal.

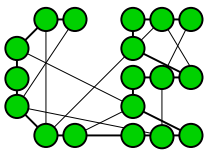
The **Kullback-Leibler** or **cross-entropy** distance is defined as:

$$\begin{aligned}\mathcal{D}(g, h) &= \mathbb{E}_g \left[\log \frac{g(\mathbf{X})}{h(\mathbf{X})} \right] \\ &= \int g(\mathbf{x}) \log g(\mathbf{x}) d\mathbf{x} - \int g(\mathbf{x}) \log h(\mathbf{x}) d\mathbf{x} .\end{aligned}$$

Note that $\mathcal{D}(g, h) \geq 0$ and $\mathcal{D}(g, h) = 0$ when $g = h$.

Shannon Entropy

$$\mathcal{H}(\mathbf{x}) = -\mathbb{E} \log f(\mathbf{X}) = - \int f(\mathbf{x}) \log f(\mathbf{x}) d\mathbf{x}$$



The Parametric CE Method

This is equivalent to solving

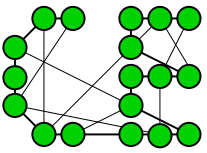
$$\max_{\mathbf{v}} D(\mathbf{v}) = \max_{\mathbf{v}} \mathbb{E}_{\mathbf{u}} [H(\mathbf{X}) \log f(\mathbf{X}; \mathbf{v})] .$$

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

$$\max_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^N I_{\{S(\mathbf{X}_i) \geq \gamma\}} \log f(\mathbf{X}_i; \mathbf{v}) ,$$

where $\mathbf{X}_1, \dots, \mathbf{X}_N$ is a random sample from $f(\cdot; \mathbf{u})$.

CE Versus Variance Minimization (VM)



Summarizing: in CE we maximize

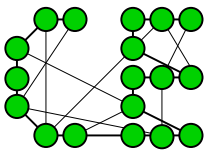
$$\max_{\mathbf{v}} \hat{\mathcal{D}}(\mathbf{v}) = \max_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \ln f(\mathbf{X}_i; \mathbf{v}) ,$$

instead of

$$\min_{\mathbf{v}} \hat{V}(\mathbf{v}) = \min_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^N H^2(\mathbf{X}_i) W(\mathbf{X}_i; \mathbf{u}, \mathbf{v}) ,$$

where $\mathbf{X}_i \sim f(\mathbf{x}; \mathbf{w})$.

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



The Inverse Transform Method

Consider estimation of $\ell = \mathbb{E}_f[H(\mathbf{X})]$, where $\mathbf{X} \sim f(\mathbf{x})$.

According to the *inverse transform* (IT) method, we can write X (for the one-dimensional case) as

$$X = F^{-1}(U),$$

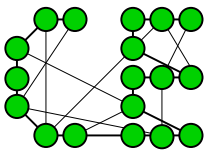
where $U \sim U(0, 1)$ and F^{-1} is the inverse of the cdf F .

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.

Combining the Inverse Transform Method with Importance Sampling

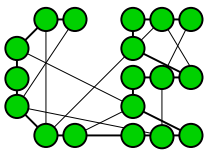


To estimate $\ell = \mathbb{E}_U[\tilde{H}(U)]$ one can use IS. As IS pdf one can take on $(0, 1)$ any pdf $h(u; \nu)$ parameterized by some reference parameter ν . An example is the $Beta(\nu, 1)$ -distribution, with density

$$h(u; \nu) = \nu u^{\nu-1}, \quad u \in (0, 1),$$

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

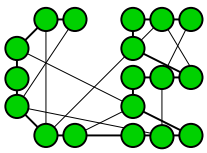
Using $Beta(1, \nu)$ as the IS pdf we can write ℓ as

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

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 ℓ is given by

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

where U_1, \dots, U_N is a random sample from $h(u; \nu)$. We call $\hat{\ell}$ the *inverse transform - likelihood ratio* (ITLR) estimator. It is generic.



Example

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

$$\hat{\ell} = N^{-1} \sum_{i=1}^N H(\lambda^{-1} (-\ln(1 - U_i))^{1/\alpha}) / h(U_i; \nu).$$



Sensitivity Analysis of Simulation Models

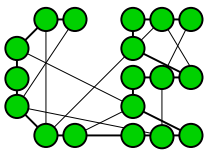
We consider the case where the expected performance is given by

$$\ell(\mathbf{u}) = \mathbb{E}_{\mathbf{u}}[H(\mathbf{X})] = \int H(\mathbf{x})f(\mathbf{x}, \mathbf{u})d\mathbf{x},$$

with $\mathbf{X} \sim f(\mathbf{x}, \mathbf{u})$ and the sensitivity is performed with respect to \mathbf{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 $\ell(\mathbf{u})$ with respect to the distributional parameter vector \mathbf{u} .

Sensitivity Analysis of Simulation Models



Consider first the case where u is scalar. Then under mild conditions the differentiation and expectation (integration) operators are interchangeable we have that

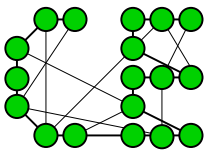
$$\begin{aligned}\frac{dl(u)}{du} &= \frac{d}{du} \int H(\mathbf{x}) f(\mathbf{x}; u) d\mathbf{x} = \int H(\mathbf{x}) \frac{df(\mathbf{x}; u)}{du} d\mathbf{x} \\ &= \int H(\mathbf{x}) \frac{\frac{df(\mathbf{x}; u)}{du}}{f(\mathbf{x}; u)} f(\mathbf{x}; u) d\mathbf{x} = \mathbb{E}_u \left[H(\mathbf{X}) \frac{d \log f(\mathbf{X}; u)}{du} \right] \\ &= \mathbb{E}_u [H(\mathbf{X}) \mathcal{S}(u; \mathbf{X})],\end{aligned}$$

where

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

is called the *score function* (SF).

Sensitivity Analysis of Simulation Models



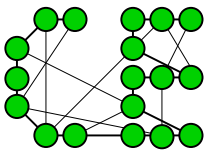
Consider next the multidimensional case. We have for the gradient and the higher order derivatives of $\ell(\mathbf{u})$

$$\nabla^k \ell(\mathbf{u}) = \mathbb{E}_{\mathbf{u}} [H(\mathbf{X}) \mathcal{S}^{(k)}(\mathbf{u}; \mathbf{X})],$$

where

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

is the k -th order score function k -th order $k = 0, 1, 2, \dots$. In particular, $\mathcal{S}^{(0)}(\mathbf{u}; \mathbf{x}) = 1$ (by definition) and $\mathcal{S}^{(1)}(\mathbf{u}; \mathbf{x}) = \mathcal{S}(\mathbf{u}; \mathbf{x}) = \nabla \log f(\mathbf{x}; \mathbf{u})$.



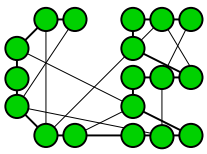
Sensitivity Analysis of Simulation Models

In general, the quantities $\nabla^k \ell(\mathbf{u})$, $k = 0, 1, \dots$, are not available analytically, since the response $\ell(\mathbf{u})$ is so. They can be estimated, however, via simulation as

$$\widehat{\nabla^k \ell(\mathbf{u})} = \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \mathcal{S}^{(k)}(\mathbf{u}; \mathbf{X}_i).$$

It is readily seen that the function $\ell(\mathbf{u})$, and *all* the sensitivities $\nabla^k \ell(\mathbf{u})$ can be estimated from a single simulation, since all of them are expressed as expectations with respect to the same pdf, $f(\mathbf{x}; \mathbf{u})$.

Sensitivity Analysis of Simulation Models: Example

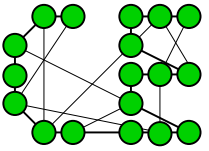


Let $H(\mathbf{X}) = X$, with $X \sim \text{Exp}(\lambda = u)$. This is a toy example since $\nabla \ell(u) = -1/u^2$. We have that $\mathcal{S}(u; x) = u^{-1} - x$, and therefore

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

is an estimator of $\nabla \ell(u)$, where X_1, \dots, X_N is a random sample from $\text{Exp}(u)$.

Sensitivity Analysis via Importance Sampling



Applying importance sampling (IS) $g(\mathbf{x})$ to $\nabla^k \ell(\mathbf{u}) = \mathbb{E}_{\mathbf{u}}[H(\mathbf{X}) \mathcal{S}^{(k)}(\mathbf{u}; \mathbf{X})]$ we obtain

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

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

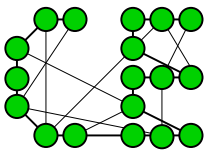
$$\widehat{\nabla^k \ell(\mathbf{u})} = \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \mathcal{S}^{(k)}(\mathbf{u}; \mathbf{X}_i) \frac{f(\mathbf{X}_i; \mathbf{u})}{g(\mathbf{X}_i)}, \quad (2)$$

where $\mathbf{X}_1, \dots, \mathbf{X}_N$ is a random sample from $g(\mathbf{x})$.



Sensitivity Analysis of Simulation Models

This means that by varying \mathbf{u} and keeping g fixed we can, in principle, estimate unbiasedly the whole *response surface* $\{\nabla^k \ell(\mathbf{u}), \mathbf{u} \in V\}$ from a *single simulation*. Often the IS distribution is chosen in the same class of distributions as the original one. That is, $g(\mathbf{x}) = f(\mathbf{x}; \mathbf{v})$, for some $\mathbf{v} \in V$.



Monte Carlo Optimization

Consider

$$\max_{\mathbf{u}} \{ \ell(\mathbf{u}) = \mathbb{E}_{\mathbf{u}}[H(\mathbf{X})] = \int H(\mathbf{x}) f(\mathbf{x}; \mathbf{u}) d\mathbf{x} \}.$$

We solve

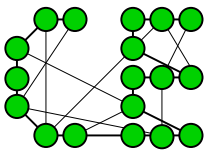
$$\nabla \ell(\mathbf{u}) = \mathbf{0} \rightarrow \int H(\mathbf{x}) \nabla \log f(\mathbf{x}; \mathbf{u}) d\mathbf{x} = \mathbf{0}.$$

For example, considering the CE program

$$\max_{\mathbf{v}} D(\mathbf{v}) = \max_{\mathbf{v}} \mathbb{E}_{\mathbf{u}} [H(\mathbf{X}) \log f(\mathbf{X}; \mathbf{v})] .$$

we obtain

$$\nabla D(\mathbf{v}) = \mathbb{E}_{\mathbf{u}} [H(\mathbf{X}) \nabla \log f(\mathbf{X}; \mathbf{v})] = \mathbf{0}.$$



Monte Carlo Optimization

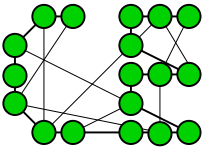
While solving the stochastic counterpart:

$$\max_{\mathbf{v}} \left\{ \frac{1}{N} \sum_{i=1}^N I_{\{S(\mathbf{x}_i) \geq \gamma\}} \log f(\mathbf{X}_i; \mathbf{v}) \right\},$$

where $\mathbf{X}_1, \dots, \mathbf{X}_N$ is a random sample from $f(\cdot; \mathbf{u})$ we would simply solve

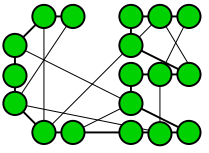
$$\frac{1}{N} \sum_{i=1}^N I_{\{S(\mathbf{x}_i) \geq \gamma\}} \nabla \log f(\mathbf{X}_i; \mathbf{v}) = \mathbf{0}$$

The Cross-Entropy Method: Applications



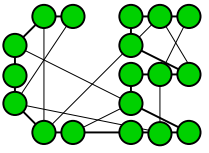
- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.

The Cross-Entropy Method: Applications



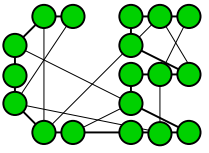
- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning

The Cross-Entropy Method: Applications



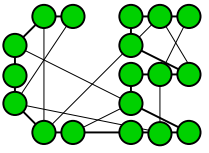
- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis

The Cross-Entropy Method: Applications



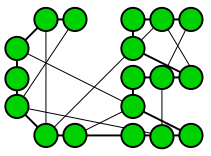
- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment

The Cross-Entropy Method: Applications



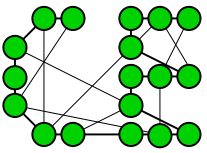
- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
- Simulation-based (noisy) Optimization, like Optimal Buffer Allocation and Optimization in Finance Engineering

The Cross-Entropy Method: Applications



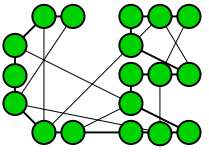
- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
- Simulation-based (noisy) Optimization, like Optimal Buffer Allocation and Optimization in Finance Engineering
- Multi-extremal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

The Cross-Entropy Method: Applications

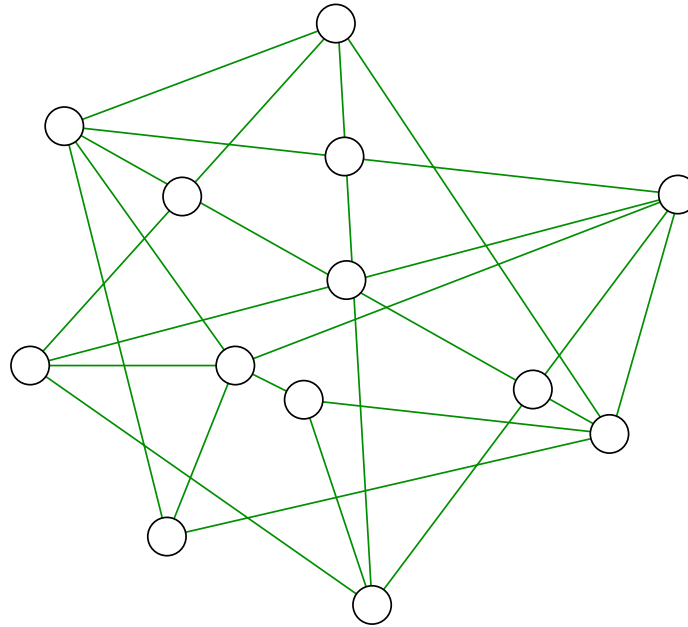


- Combinatorial Optimization, like TSP, Maximal Cut, Scheduling and Production Lines.
- Machine Learning
- Pattern Recognition, Clustering and Image Analysis
- DNA Sequence Alignment
- Simulation-based (noisy) Optimization, like Optimal Buffer Allocation and Optimization in Finance Engineering
- Multi-extremal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

Combinatorial Optimization: A Coloring Problem

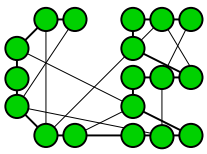


We wish to color the nodes white and black.

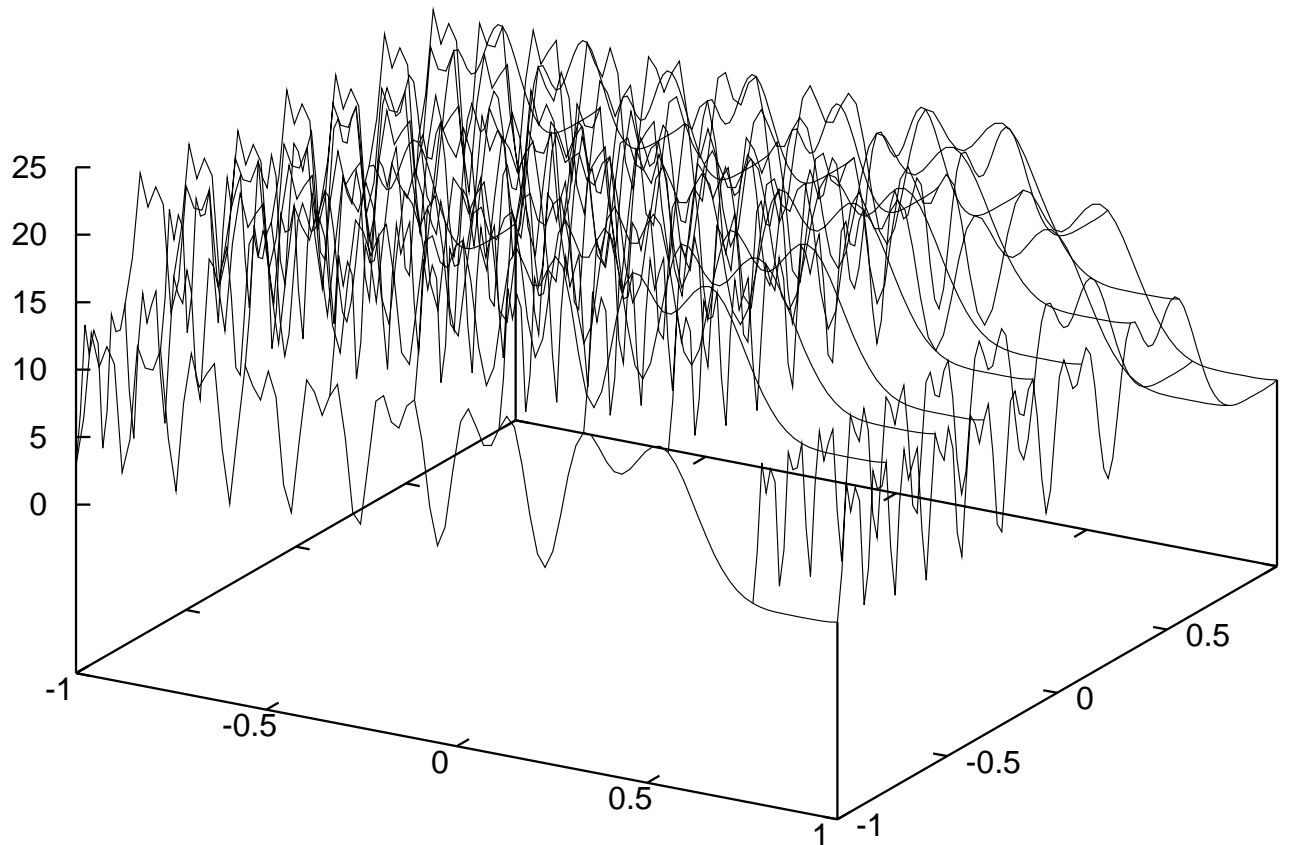


How should we color so that the total number of links **between** the two groups is maximized? This problem is known as *Maximal Cut* problem.

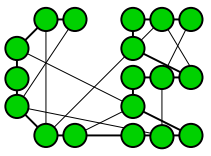
Continuous Optimization: A Multi-extremal function



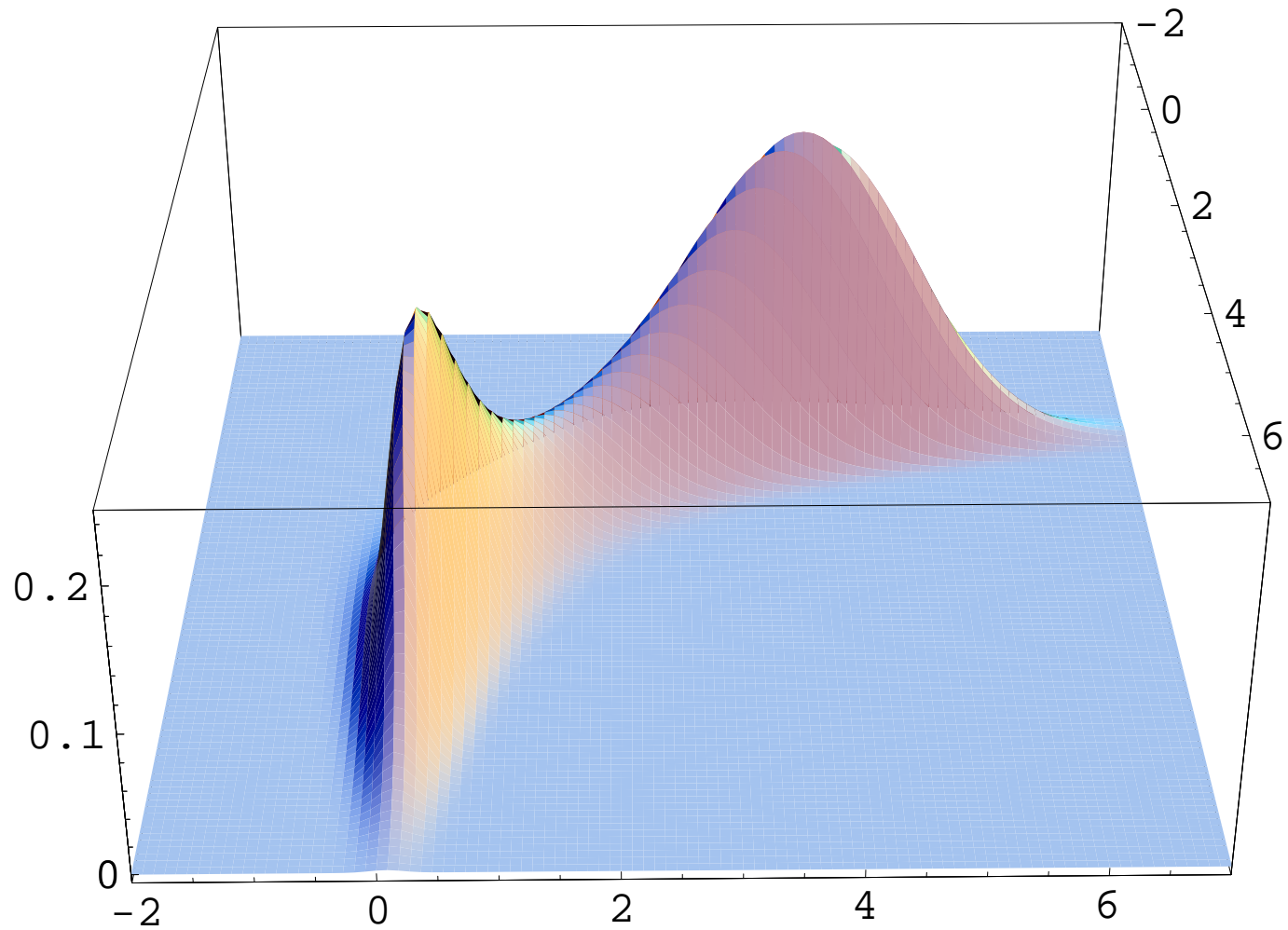
This is the trigonometric function.

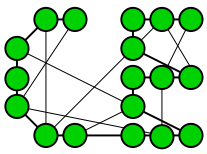


What is its global maximum, and where is it attained?

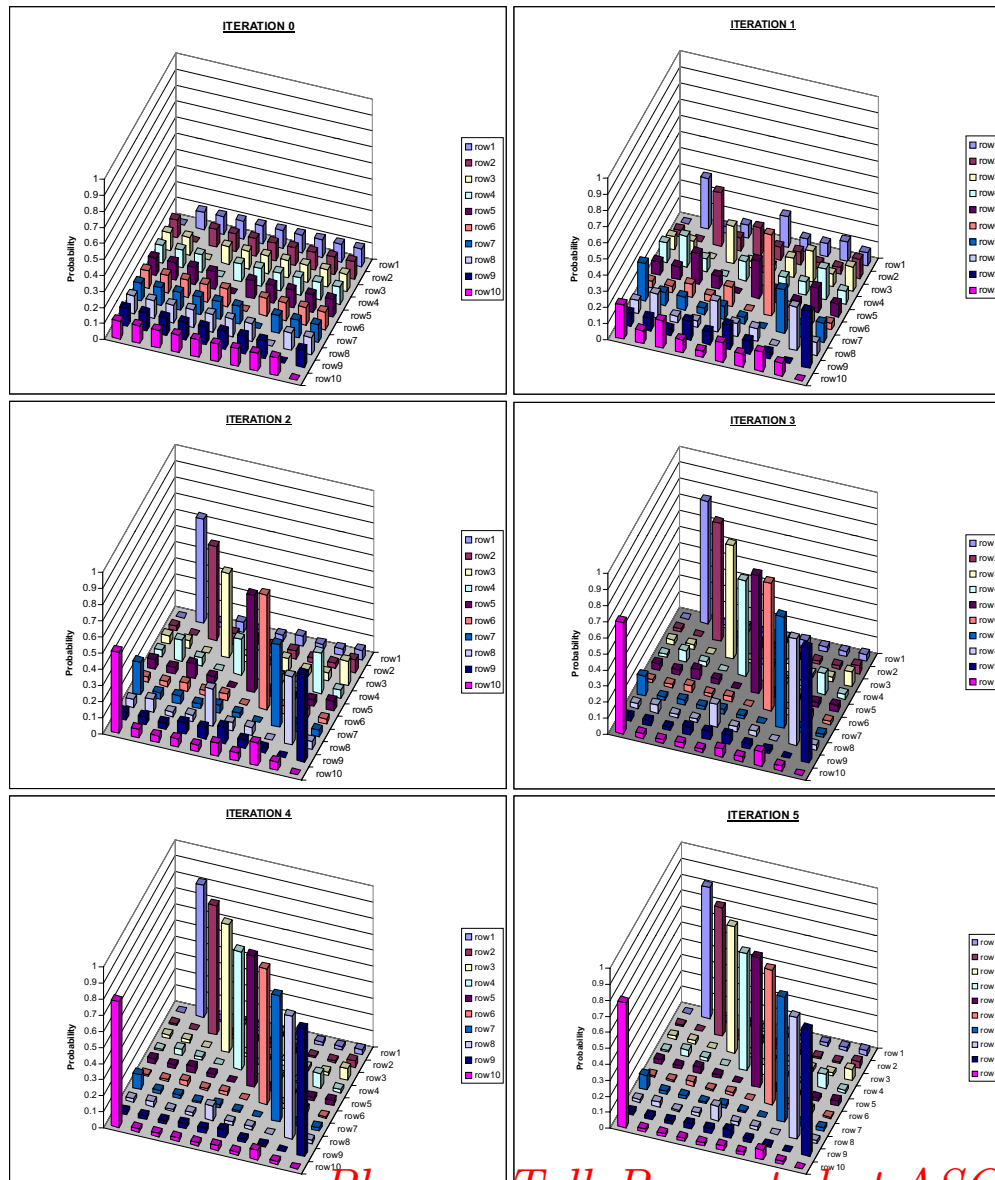


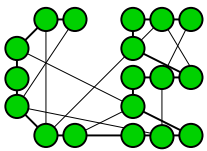
Another Multi-extremal function



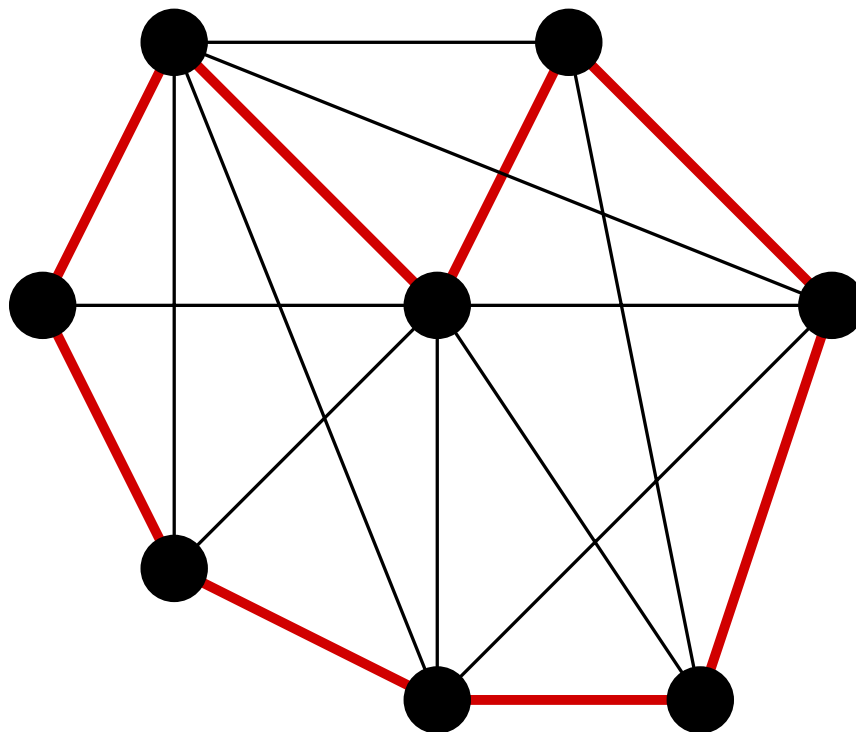


Combinatorial Optimization: TSP

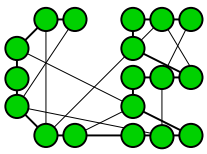




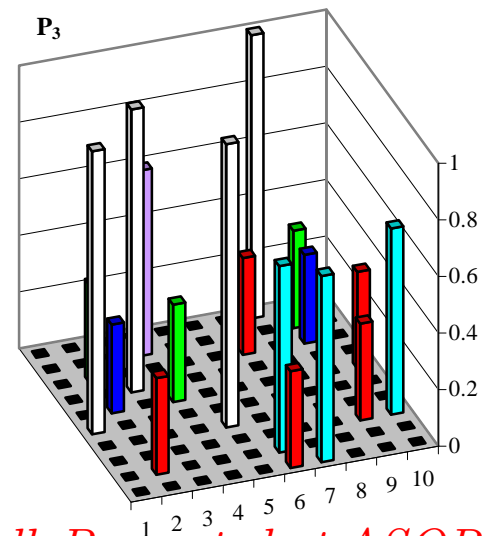
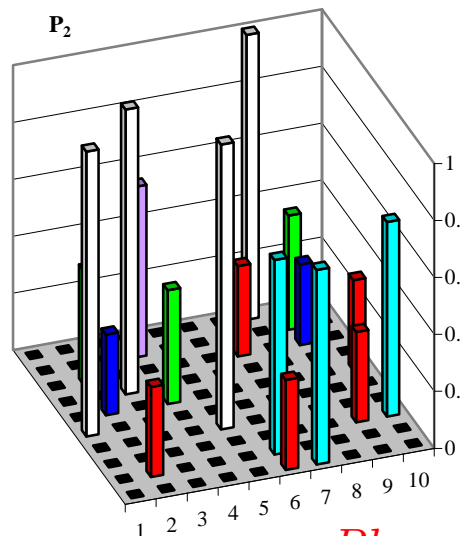
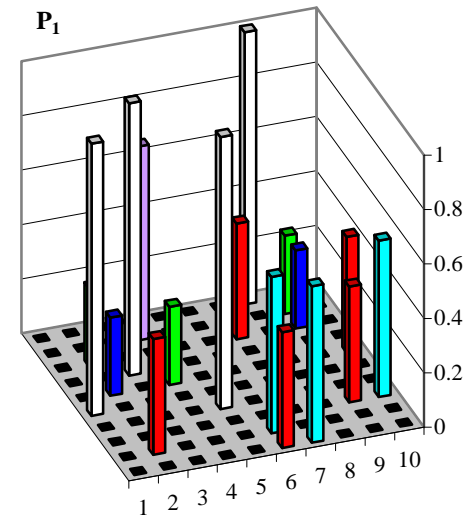
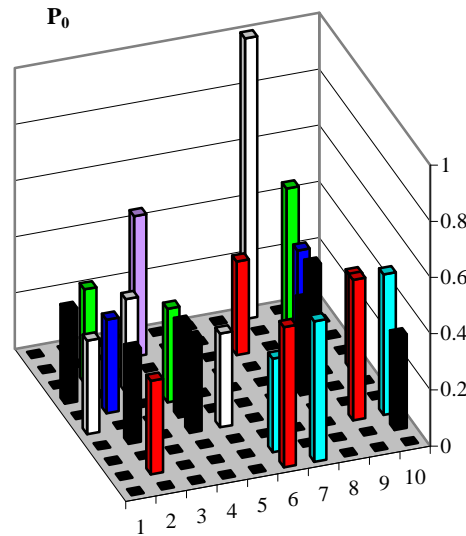
Counting Hamiltonian Cycles

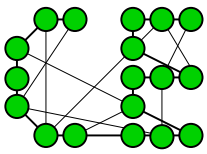


How many Hamiltonian cycles does this graph have?



Calculating the Number of HC's

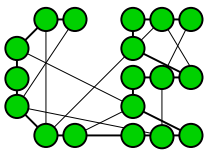




Hamiltonian Cycles: Large Example.

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

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



Introduction

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

1. Estimation:

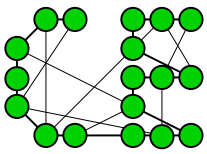
$$\text{Estimate } \ell = \mathbb{E}[H(\mathbf{X})]$$

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

H : function on \mathcal{X} .

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

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



Introduction

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

1. Estimation:

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

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

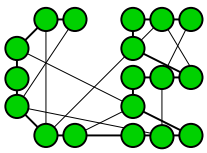
H : function on \mathcal{X} .

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

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

2. Optimization:

$$\text{Determine } \max_{\mathbf{x} \in \mathcal{X}} S(\mathbf{x})$$



The Simplest Example

Consider the problem of estimating

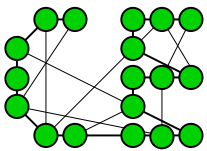
$$\ell = \mathbb{E}_u [I_{\{X \geq \gamma\}}] = \mathbb{P}(X \geq \gamma),$$

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

$$\begin{aligned} D(v) &= \mathbb{E}_u [I_{\{X \geq \gamma\}} W(X; u, w) \log f(X; v)] \\ &= \mathbb{E}_u \left[I_{\{X \geq \gamma\}} W(X; u, w) \left(-\log(v) - \frac{X}{v} \right) \right]. \end{aligned}$$



The Simplest Example

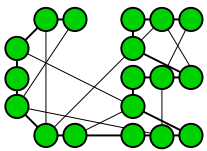
Setting $D'(v) = 0$ we find

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

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

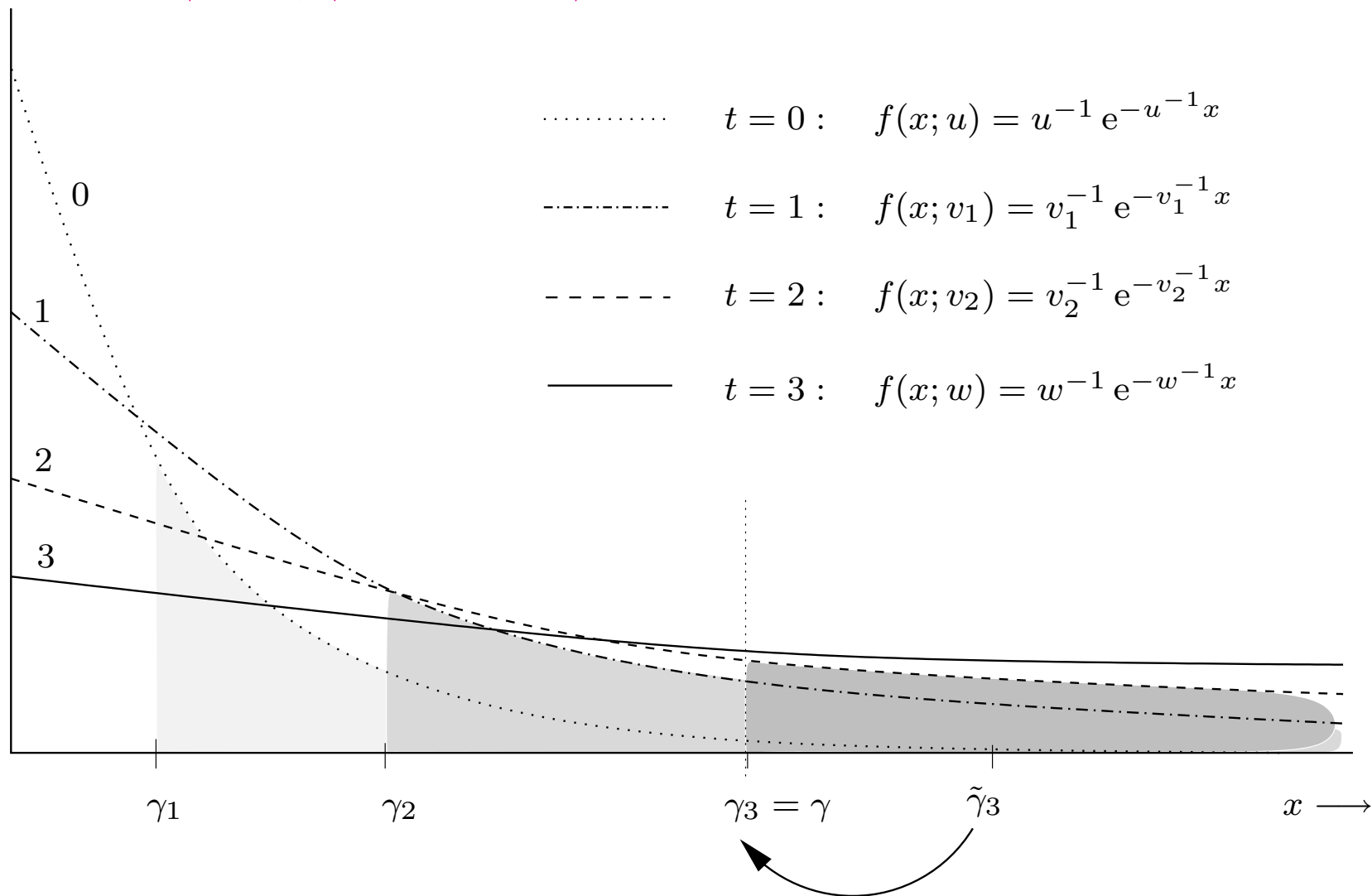
$$\hat{v} = \frac{\sum_{i=1}^N I_{\{X_i \geq \gamma\}} W(X_i; u, w) X_i}{\sum_{i=1}^N I_{\{X_i \geq \gamma\}} W(X_i; u, w)},$$

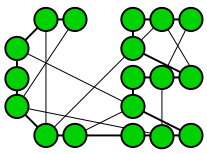
with $X_1, \dots, X_N \sim f(\cdot; w)$.



General Algorithm

Generate $\gamma_1, v_1, \gamma_2, v_2, \dots, \gamma_T, v_T$.





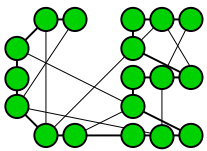
Generic CE Algorithm

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

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

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

$$\hat{v}_{t,j} = \frac{\sum_{i=1}^N I_{\{S(\mathbf{x}_i) \geq \hat{\gamma}_t\}} W(\mathbf{X}_i; \mathbf{u}, \hat{\mathbf{v}}_{t-1}) X_{ij}}{\sum_{i=1}^N I_{\{S(\mathbf{x}_i) \geq \hat{\gamma}_t\}} W(\mathbf{X}_i; \mathbf{u}, \hat{\mathbf{v}}_{t-1})} .$$

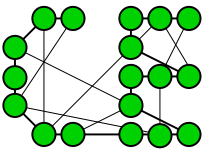


Combinatorial Optimization

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

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

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



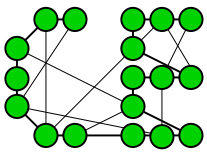
General CE Procedure

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

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

and involves the following iterative steps:

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



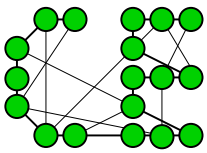
General CE Procedure

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

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

and involves the following iterative steps:

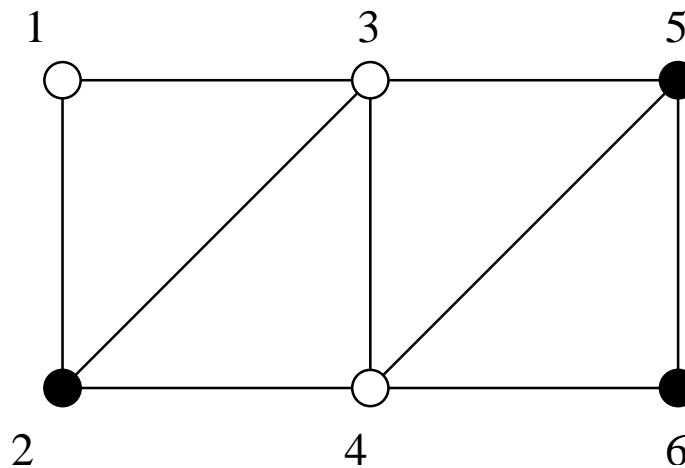
- Formulate a parameterized random mechanism to *generate* the objects $\mathbf{x} \in \mathcal{X}$.
- Give the *updating formulas* for the parameters of the random mechanism (obtained via Cross-Entropy minimization), in order to produce a better sample in the next iteration.

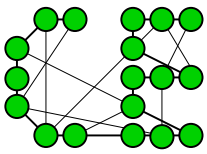


Max-Cut Example

Consider a weighted graph G with node set $V = \{1, \dots, n\}$. 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





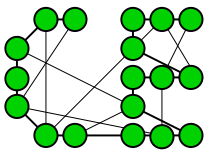
The Max-Cut problem

Cost matrix:

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

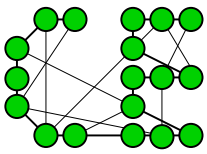
$\{V_1, V_2\} = \{\{1, 3, 4\}, \{2, 5, 6\}\}$ is a possible **cut**. The **cost** of the cut is

$$c_{12} + c_{32} + c_{35} + c_{42} + c_{45} + c_{46}.$$



Generation and Updating Formulas

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



Generation and Updating Formulas

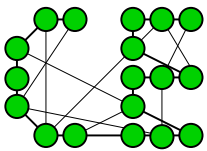
Generation of cut vectors: The most natural and easiest way to generate the cut vectors is to let X_2, \dots, X_n be independent Bernoulli random variables with success probabilities p_2, \dots, p_n .

Updating formulas:

$$\hat{p}_{t,j} = \frac{\sum_{i=1}^N I_{\{S(\mathbf{x}_i) \geq \hat{\gamma}_t\}} I_{\{X_{ij}=1\}}}{\sum_{i=1}^N I_{\{S(\mathbf{x}_i) \geq \hat{\gamma}_t\}}} = \frac{\sum_{\mathbf{x}_i \in \mathcal{E}_t} X_{ij}}{|\mathcal{E}_t|}, \quad j = 2, \dots, n$$

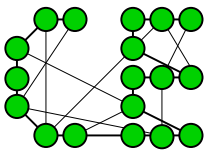
where $|\mathcal{E}_t| = \rho N$, the number of **elite samples**.

Note that the likelihood term is missing.



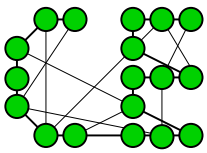
CE Optimization Algorithm

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



CE Optimization Algorithm

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

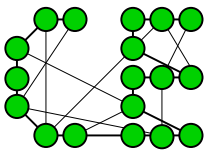


CE Optimization Algorithm

- 1 Start with $\hat{p}_0 = (1/2, \dots, 1/2)$. Let $t := 1$.
- 2 **Update $\hat{\gamma}_t$** : Draw $\mathbf{X}_1, \dots, \mathbf{X}_N$ 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

$$\hat{p}_{t,j} = \frac{\sum_{\mathbf{x}_i \in \mathcal{E}_t} X_{ij}}{|\mathcal{E}_t|},$$

$j = 1, \dots, n$, where $\mathbf{X}_i = (X_{i1}, \dots, X_{in})$, and increase t by 1.



CE Optimization Algorithm

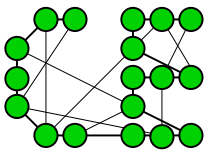
- 1 Start with $\hat{p}_0 = (1/2, \dots, 1/2)$. Let $t := 1$.
- 2 **Update $\hat{\gamma}_t$** : Draw $\mathbf{X}_1, \dots, \mathbf{X}_N$ from $\text{Ber}(\hat{p}_t)$. Let $\hat{\gamma}_t$ be the worst performance of the $\rho \times 100\%$ best performances.

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

$$\hat{p}_{t,j} = \frac{\sum_{\mathbf{x}_i \in \mathcal{E}_t} X_{ij}}{|\mathcal{E}_t|},$$

$j = 1, \dots, n$, where $\mathbf{X}_i = (X_{i1}, \dots, X_{in})$, and increase t by 1.

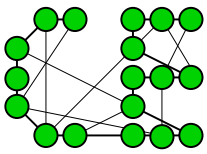
- 4 If the stopping criterion is met, then stop; otherwise set $t := t + 1$ and reiterate from step 2.



A Knapsack Problem

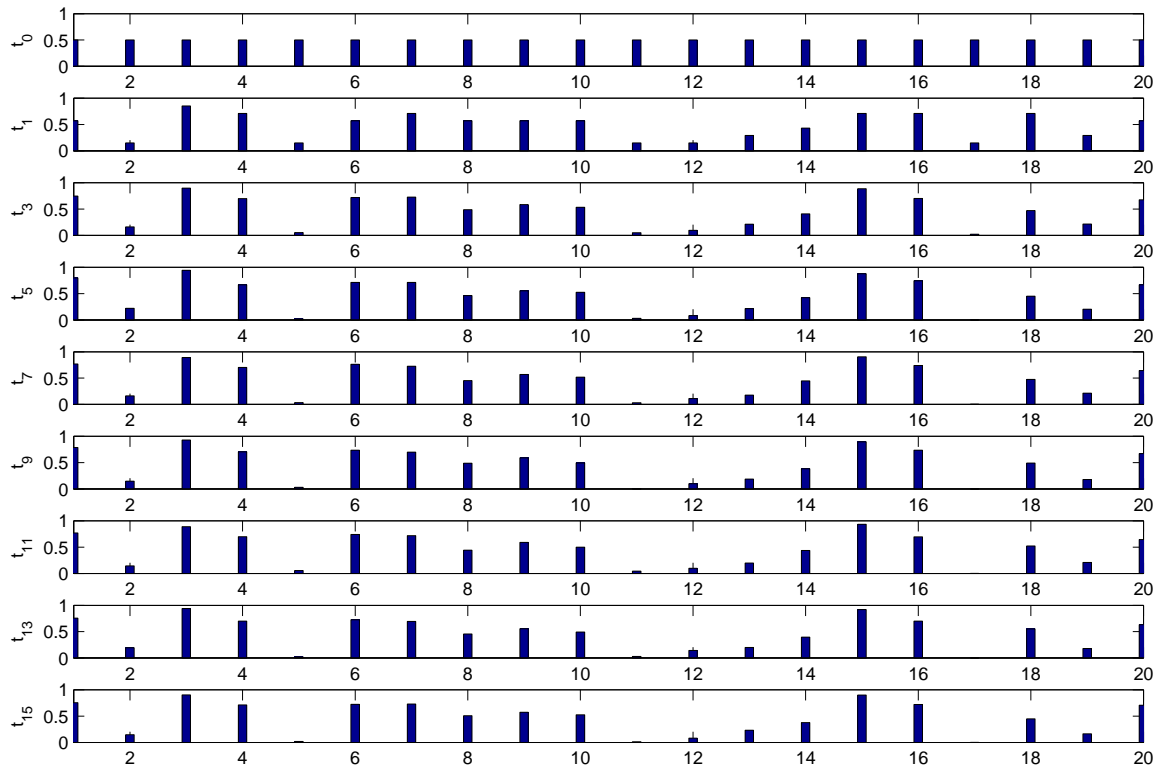
Performance of the CE Algorithm for the knapsack problem with the instance matrix $A = (20 \times 11)$ and $N = 10,000$. This problem was taken from the website <http://elib.zib.de>. Using full enumeration we found that the total number of multiple extrema is 612.

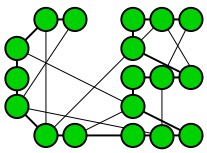
t	Mean	Max	Min	PV	RE	S	m	RD_c
0	639.6	943.7	419.4	0.00	0.225	6.93	10	55.73
1	619.2	697.6	564.8	0.03	0.072	5.78	11	0.02
2	630.8	706.5	557.0	0.07	0.059	5.18	11	0.03
3	628.1	698.1	533.2	0.08	0.083	4.95	11	0.03
4	573.7	671.2	504.9	0.09	0.083	4.88	11	0.06
5	599.3	719.6	525.7	0.09	0.100	4.72	11	0.03
6	576.9	646.4	508.0	0.09	0.071	4.76	11	0.06



A Knapsack Problem

A typical dynamics of the CE Algorithm for the knapsack problem with the instance matrix $A = (20 \times 11)$ and $N = 10,000$.





The Screening Method

Consider estimation of

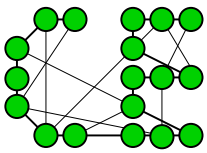
$$\ell = \mathbb{P}_{\mathbf{u}}(S(\mathbf{X}) \geq \gamma) = \mathbb{E}_{\mathbf{u}} [I_{\{S(\mathbf{X}) \geq \gamma\}}]$$

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

$$\hat{\ell} = \frac{1}{N} \sum_{k=1}^N I_{\{S(\mathbf{X}_k) \geq \gamma\}} W(\mathbf{X}_k; \mathbf{u}, \mathbf{v}),$$

where $\mathbf{X}_1, \dots, \mathbf{X}_N$ is a random sample from $f(\mathbf{x}; \mathbf{v})$, and

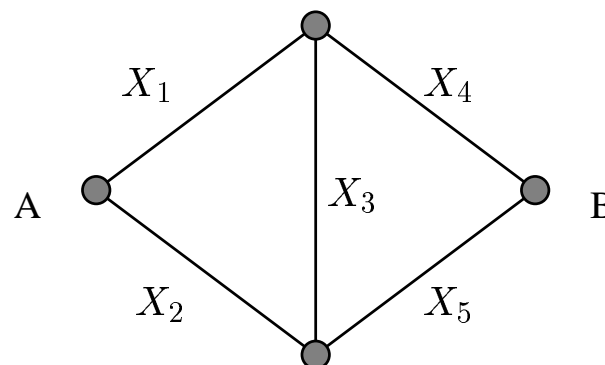
$W(\mathbf{X}_k; \mathbf{u}, \mathbf{v}) = f(\mathbf{X}_k; \mathbf{u}) / f(\mathbf{X}_k; \mathbf{v})$ is the likelihood ratio.

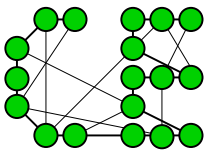


Example: Stochastic Shortest Path

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

$$S(\mathbf{X}) = \min\{X_1 + X_4, X_1 + X_3 + X_5, X_2 + X_5, X_2 + X_3 + X_4\}.$$



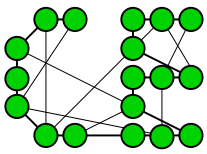


The Screening Method

The table below displays the performance of the CE algorithm.

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

t	$\hat{\gamma}_t$	$\hat{\mathbf{v}}_t$				
0		1.0000	1.0000	0.3000	0.2000	0.1000
1	1.1656	1.9805	2.0078	0.3256	0.2487	0.1249
2	2.1545	2.8575	3.0006	0.2554	0.2122	0.0908
3	3.1116	3.7813	4.0858	0.3017	0.1963	0.0764
4	4.6290	5.2803	5.6542	0.2510	0.1951	0.0588
5	6.0000	6.7950	6.7094	0.2882	0.1512	0.1360

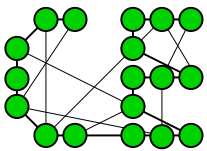


SUMMARY

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