



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)  
[Donate to Wikipedia](#)  
[Wikipedia store](#)

Interaction

[Help](#)  
[About Wikipedia](#)  
[Community portal](#)  
[Recent changes](#)  
[Contact page](#)

Tools

[What links here](#)  
[Related changes](#)  
[Upload file](#)  
[Special pages](#)  
[Permanent link](#)  
[Page information](#)  
[Wikidata item](#)  
[Cite this page](#)

Print/export

[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)

Languages

[Français](#)  
[ไทย](#)  
[Українська](#)

 [Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

# Cross-entropy method

From Wikipedia, the free encyclopedia



This article includes a [list of references](#), related reading or [external links](#), **but its sources remain unclear because it lacks [inline citations](#)**.

Please [improve](#) this article by introducing more precise citations.  
(*September 2013*)



This article **needs additional citations for [verification](#)**. Please help [improve this article](#) by [adding citations to reliable sources](#). Unsourced material may be challenged and removed. (*September 2013*)

The **cross-entropy (CE) method** attributed to [Reuven Rubinstein](#) is a general [Monte Carlo](#) approach to [combinatorial](#) and [continuous](#) multi-extremal [optimization](#) and [importance sampling](#). The method originated from the field of *rare event simulation*, where very small probabilities need to be accurately estimated, for example in network reliability analysis, queueing models, or performance analysis of telecommunication systems. The CE method can be applied to static and noisy combinatorial optimization problems such as the [traveling salesman problem](#), the [quadratic assignment problem](#), [DNA sequence alignment](#), the [max-cut](#) problem and the buffer allocation problem, as well as continuous [global optimization](#) problems with many local [extrema](#).

In a nutshell the CE method consists of two phases:

1. Generate a random data sample (trajectories, vectors, etc.) according to a specified mechanism.
2. Update the parameters of the random mechanism based on the data to produce a "better" sample in the next iteration. This step involves minimizing the *cross-entropy* or [Kullback–Leibler divergence](#).

## Contents [\[hide\]](#)

- 1 Estimation via importance sampling
- 2 Generic CE algorithm
- 3 Continuous optimization—example
  - 3.1 Pseudo-code
- 4 Related methods
- 5 See also
- 6 References
- 7 External links

## Estimation via importance sampling [\[edit\]](#)

Consider the general problem of estimating the quantity  $\ell = \mathbb{E}_{\mathbf{u}}[H(\mathbf{X})] = \int H(\mathbf{x}) f(\mathbf{x}; \mathbf{u}) \, d\mathbf{x}$ , where  $H$  is some *performance function* and  $f(\mathbf{x}; \mathbf{u})$  is a member of some [parametric family](#) of distributions. Using [importance sampling](#) this quantity can be estimated as  $\hat{\ell} = \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \frac{f(\mathbf{X}_i; \mathbf{u})}{g(\mathbf{X}_i)}$ , where

$\mathbf{X}_1, \dots, \mathbf{X}_N$  is a random sample from  $g$ . For positive  $H$ , the theoretically *optimal* importance sampling [density](#) (pdf) is given by  $g^*(\mathbf{x}) = H(\mathbf{x})f(\mathbf{x}; \mathbf{u})/\ell$ . This, however, depends on the unknown  $\ell$ . The CE method aims to approximate the optimal PDF by adaptively selecting members of the parametric family that are closest (in the [Kullback–Leibler](#) sense) to the optimal PDF  $g^*$ .

## Generic CE algorithm [\[edit\]](#)

1. Choose initial parameter vector  $\mathbf{v}^{(0)}$ ; set  $t = 1$ .
2. Generate a random sample  $\mathbf{X}_1, \dots, \mathbf{X}_N$  from  $f(\cdot; \mathbf{v}^{(t-1)})$
3. Solve for  $\mathbf{v}^{(t)}$ , where

$$\mathbf{v}^{(t)} = \underset{\mathbf{v}}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \frac{f(\mathbf{X}_i; \mathbf{u})}{f(\mathbf{X}_i; \mathbf{v}^{(t-1)})} \log f(\mathbf{X}_i; \mathbf{v})$$

4. If convergence is reached then **stop**; otherwise, increase  $t$  by 1 and reiterate from step 2.

In several cases, the solution to step 3 can be found *analytically*. Situations in which this occurs are

- When  $f$  belongs to the [natural exponential family](#)
- When  $f$  is [discrete](#) with finite [support](#)
- When  $H(\mathbf{X}) = \mathbb{I}_{\{\mathbf{x} \in A\}}$  and  $f(\mathbf{X}_i; \mathbf{u}) = f(\mathbf{X}_i; \mathbf{v}^{(t-1)})$ , then  $\mathbf{v}^{(t)}$  corresponds to the [maximum likelihood estimator](#) based on those  $\mathbf{X}_k \in A$ .

## Continuous optimization—example [\[edit\]](#)

The same CE algorithm can be used for optimization, rather than estimation. Suppose the problem is to maximize some function  $S(\mathbf{x})$ , for example,  $S(x) = e^{-(x-2)^2} + 0.8 e^{-(x+2)^2}$ . To apply CE, one considers first the *associated stochastic problem* of estimating  $\mathbb{P}_{\boldsymbol{\theta}}(S(\mathbf{X}) \geq \gamma)$  for a given *level*  $\gamma$ , and parametric family  $\{f(\cdot; \boldsymbol{\theta})\}$ , for example the 1-dimensional [Gaussian distribution](#), parameterized by its mean  $\mu_t$  and variance  $\sigma_t^2$  (so  $\boldsymbol{\theta} = (\mu, \sigma^2)$  here). Hence, for a given  $\gamma$ , the goal is to find  $\boldsymbol{\theta}$  so that  $D_{\text{KL}}(\mathbb{I}_{\{S(\mathbf{x}) \geq \gamma\}} \| f_{\boldsymbol{\theta}})$  is minimized. This is done by solving the sample version (stochastic counterpart) of the KL divergence minimization problem, as in step 3 above. It turns out that parameters that minimize the stochastic counterpart for this choice of target distribution and parametric family are the sample mean and sample variance corresponding to the *elite samples*, which are those samples that have objective function value  $\geq \gamma$ . The worst of the elite samples is then used as the level parameter for the next iteration. This yields the following randomized algorithm that happens to coincide with the so-called Estimation of Multivariate Normal Algorithm (EMNA), an [estimation of distribution algorithm](#).

### Pseudo-code [\[edit\]](#)

```
1. mu:=-6; sigma2:=100; t:=0; maxits=100;      // Initialize parameters
2. N:=100; Ne:=10;                             //
3. while t < maxits and sigma2 > epsilon        // While maxits not exceeded and not
   converged
4. X = SampleGaussian(mu,sigma2,N);            // Obtain N samples from current
   sampling distribution
5. S = exp(-(X-2)^2) + 0.8 exp(-(X+2)^2);      // Evaluate objective function at
   sampled points
6. X = sort(X,S);                              // Sort X by objective function values
   (in descending order)
7. mu = mean(X(1:Ne)); sigma2=var(X(1:Ne));    // Update parameters of sampling
   distribution
8. t = t+1;                                    // Increment iteration counter
9. return mu                                  // Return mean of final sampling
   distribution as solution
```


## Related methods [\[edit\]](#)

- [Simulated annealing](#)
- [Genetic algorithms](#)
- [Harmony search](#)
- [Estimation of distribution algorithm](#)
- [Tabu search](#)

## See also [\[edit\]](#)

- [Cross entropy](#)
- [Kullback–Leibler divergence](#)
- [Randomized algorithm](#)
- [Importance sampling](#)

## References [\[edit\]](#)

- De Boer, P.-T., Kroese, D.P, Mannor, S. and Rubinstein, R.Y. (2005). A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, **134** (1), 19–67.[\[1\]](#) 
- Rubinstein, R.Y. (1997). Optimization of Computer simulation Models with Rare Events, *European Journal of Operations Research*, **99**, 89–112.

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

## External links [\[edit\]](#)

- [Homepage for the CE method](#) [↗](#)
- [CEoptim R package](#) [↗](#)

Categories: [Heuristics](#) | [Optimization algorithms and methods](#) | [Monte Carlo methods](#) | [Machine learning](#)

This page was last modified on 14 April 2015, at 01:17.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

