Cross-Entropy Method

Dirk P. Kroese
School of Mathematics and Physics
The University of Queensland
Brisbane 4072, Australia
kroese@maths.uq.edu.au

Abstract: The cross-entropy method is a recent versatile Monte Carlo technique. This article provides a brief introduction to the cross-entropy method and discusses how it can be used for rare-event probability estimation and for solving combinatorial, continuous, constrained and noisy optimization problems. A comprehensive list of references on cross-entropy methods and applications is included.

Keywords: cross-entropy, Kullback-Leibler divergence, rare events, importance sampling, stochastic search.

The *cross-entropy (CE) method* is a recent generic Monte Carlo technique for solving complicated simulation and optimization problems. The approach was introduced by R.Y. Rubinstein in [41, 42], extending his earlier work on variance minimization methods for rare-event probability estimation [40].

The CE method can be applied to two types of problem:

1. **Estimation**: Estimate $\ell = \mathbb{E}[H(\mathbf{X})]$, where $\mathbf{X}$ is a random variable or vector taking values in some set $\mathscr{X}$ and H is function on $\mathscr{X}$. An important special case is the estimation of a probability $\ell = \mathbb{P}(S(\mathbf{X}) \geqslant \gamma)$, where S is another function on $\mathscr{X}$.

2. **Optimization**: Optimize (that is, maximize or minimize) $S(\mathbf{x})$ over all $\mathbf{x} \in \mathscr{X}$, where S is some objective function on $\mathscr{X}$. S can be either a known or a *noisy* function. In the latter case the objective function needs to be estimated, e.g., via simulation.

In the estimation setting, the CE method can be viewed as an adaptive *importance sampling* procedure that uses the *cross-entropy or Kullback-Leibler divergence* as a measure of closeness between two sampling distributions, as is explained further in Section 1. In the optimization setting, the optimization problem is first translated into a rare-event estimation problem, and then the CE method for estimation is used as an adaptive algorithm to locate the optimum, as is explained further in Section 2.

An easy tutorial on the CE method is given in [15]. A more comprehensive treatment can be found in [45]; see also [46, Chapter 8]. The CE method homepage can be found at `www.cemethod.org` .

The CE method has been successfully applied to a diverse range of estimation and optimization problems, including buffer allocation [1], queueing models of telecommunication systems [14, 16], optimal control of HIV/AIDS spread [48, 49], signal detection [30], combinatorial auctions [9], DNA sequence alignment [24, 38], scheduling and vehicle routing [3, 8, 11, 20, 23, 53], neural and reinforcement learning [31, 32, 34, 52, 54], project management [12], rare-event simulation with light- and heavy-tail distributions [2, 10, 21, 28], clustering analysis [4, 5, 29]. Applications to classical combinatorial optimization problems including the max-cut, traveling salesman, and Hamiltonian cycle

problems are given in [7, 17, 42, 43, 44]. Various CE estimation and noisy optimization problems for reliability systems and network design can be found in [6, 22, 25, 26, 35, 36, 37, 39]. Parallel implementations of the CE method are discussed in [18, 19], and recent generalizations and advances are explored in [51].

1 The Cross-Entropy Method for Estimation

Consider the estimation of

$$\ell = \mathbb{E}_f[H(\mathbf{X})] = \int H(\mathbf{x})\, f(\mathbf{x})\, \mathrm{d}\mathbf{x}\ , \tag{1}$$

where H is the *sample performance function* and f is the probability density of the random variable (vector) $\mathbf{X}$. (For notational convenience it is assumed that $\mathbf{X}$ is a continuous random variable; if $\mathbf{X}$ is a discrete random variable, simply replace the integral in (1) by a sum.) Let g be another probability density such that for all $\mathbf{x}$, $g(\mathbf{x}) = 0$ implies that $H(\mathbf{x})\, f(\mathbf{x}) = 0$. Using the probability density g, we can represent ℓ as

$$\ell = \int H(\mathbf{x})\, \frac{f(\mathbf{x})}{g(\mathbf{x})}\, g(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}_g \left[H(\mathbf{X})\, \frac{f(\mathbf{X})}{g(\mathbf{X})} \right]\ . \tag{2}$$

Consequently, if $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are independent random vectors, each with probability density g, then

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k)\, \frac{f(\mathbf{X}_k)}{g(\mathbf{X}_k)} \tag{3}$$

is an unbiased estimator of ℓ. Such an estimator is called an *importance sampling estimator*. The optimal importance sampling probability density is given by $g^*(\mathbf{x}) \propto |H(\mathbf{x})| f(\mathbf{x})$ (see, e.g., [46, page 132]), which in general is difficult to obtain. The idea of the CE method is to choose the importance sampling density g in a specified class of densities such that the *cross-entropy* or *Kullback-Leibler divergence* between the optimal importance sampling density g^* and g is minimal. The Kullback-Leibler divergence between two probability densities g and h is given by

$$\begin{aligned} \mathcal{D}(g, h) &= \mathbb{E}_g \left[\ln \frac{g(\mathbf{X})}{h(\mathbf{X})} \right] = \int g(\mathbf{x}) \ln \frac{g(\mathbf{x})}{h(\mathbf{x})}\, \mathrm{d}\mathbf{x} \\ &= \int g(\mathbf{x}) \ln g(\mathbf{x})\, \mathrm{d}\mathbf{x} - \int g(\mathbf{x}) \ln h(\mathbf{x})\, \mathrm{d}\mathbf{x}\ . \end{aligned} \tag{4}$$

In most cases of interest the sample performance function H is non-negative, and the "nominal" probability density f is parameterized by a finite-dimensional vector $\mathbf{u}$; that is, $f(\mathbf{x}) = f(\mathbf{x}; \mathbf{u})$. It is then customary to choose the importance sampling probability density g in the *same* family of probability densities; thus, $g(\mathbf{x}) = f(\mathbf{x}; \mathbf{v})$ for some *reference parameter* $\mathbf{v}$. The CE minimization procedure then reduces to finding an optimal reference parameter vector, $\mathbf{v}^*$ say, by cross-entropy minimization. This $\mathbf{v}^*$ turns out to be the solution to the maximization

problem $\max_{\mathbf{v}} \int H(\mathbf{x}) f(\mathbf{x};\mathbf{u}) \ln f(\mathbf{x};\mathbf{v}) \mathrm{d}\mathbf{x}$, which in turn can be estimated via simulation by solving with respect to $\mathbf{v}$, the stochastic counterpart program

$$\max_{\mathbf{v}} \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k) \frac{f(\mathbf{X}_k;\mathbf{u})}{f(\mathbf{X}_k;\mathbf{w})} \ln f(\mathbf{X}_k;\mathbf{v}) \,, \tag{5}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a random sample from $f(\cdot;\mathbf{w})$, for any reference parameter $\mathbf{w}$. The maximization (5) can often be solved *analytically*, in particular when the class of sampling distributions forms an *exponential family*; see, for example, [46, pages 319–320]. Indeed, analytical updating formulas can be found whenever explicit expressions for the *maximal likelihood estimators* of the parameters can be found, cf. [15, page 36].

Often $\ell = \mathbb{P}(S(\mathbf{X}) \geqslant \gamma)$, for some performance function S and *level* γ, in which case $H(\mathbf{x})$ takes the form of an *indicator function*: $H(\mathbf{x}) = I_{\{S(\mathbf{X}) \geqslant \gamma\}}$; that is, $H(\mathbf{x}) = 1$ if $S(\mathbf{x}) \geqslant \gamma$, and 0 otherwise. A complication in solving (5) occurs when ℓ is a *rare-event probability*; that is, a very small probability (say less than 10^{-5}). Then, for moderate sample size N most or all of the values $H(\mathbf{X}_k)$ in (5) are zero, and the maximization problem becomes useless. In that case a *multi-level* CE procedure is used, where a sequence of reference parameters and levels is constructed with the goal that the first converges to $\mathbf{v}^*$ and the second to γ. This leads to the following algorithm; see, e.g., [46, page 238].

Algorithm 1.1 (CE Algorithm for Rare-Event Estimation)

1. *Define* $\widehat{\mathbf{v}}_0 = \mathbf{u}$. *Let* $N^{\mathrm{e}} = \lceil \varrho N \rceil$. *Set* $t = 1$ *(iteration counter).*

2. *Generate a random sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *according to the probability density* $f(\cdot;\widehat{\mathbf{v}}_{t-1})$. *Calculate the performances* $S(\mathbf{X}_i)$ *for all* i, *and order them from smallest to largest,* $S_{(1)} \leqslant \ldots \leqslant S_{(N)}$. *Let* $\widehat{\gamma}_t$ *be the sample* $(1-\varrho)$*-quantile of performances; that is,* $\widehat{\gamma}_t = S_{(N-N^{\mathrm{e}}+1)}$. *If* $\widehat{\gamma}_t > \gamma$, *reset* $\widehat{\gamma}_t$ *to* γ.

3. *Use the* **same** *sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *to solve the stochastic program* (5), *with* $\mathbf{w} = \widehat{\mathbf{v}}_{t-1}$. *Denote the solution by* $\widehat{\mathbf{v}}_t$.

4. *If* $\widehat{\gamma}_t < \gamma$, *set* $t = t + 1$ *and reiterate from Step 2; otherwise, proceed with Step 5.*

5. *Let* T *be the final iteration counter. Generate a sample* $\mathbf{X}_1, \ldots, \mathbf{X}_{N_1}$ *according to the probability density* $f(\cdot;\widehat{\mathbf{v}}_T)$ *and estimate* ℓ *via importance sampling, as in* (3).

Apart from specifying the family of sampling probability densities, the initial vector $\widehat{\mathbf{v}}_0$, the sample size N and the rarity parameter ϱ (typically between 0.01 and 0.1), the algorithm is completely self-tuning. The sample size N for determining a good reference parameter can usually be chosen much smaller than the sample size N_1 for the final importance sampling estimation, say $N =$

1000 versus $N_1 = 100,000$. Under certain technical conditions the deterministic version of Algorithm 1.1 is guaranteed to terminate (reach level γ) provided that ϱ is chosen small enough; see Section 3.5 of [45].

2 The Cross-Entropy Method for Optimization

Let $\mathscr{X}$ be an arbitrary set of *states* and let S be a real-valued performance function on $\mathscr{X}$. Suppose we wish to find the maximum of S over $\mathscr{X}$, and the corresponding state $\mathbf{x}^*$ at which this maximum is attained (assuming for simplicity that there is only one such state). Denote the maximum by γ^*, we thus have

$$S(\mathbf{x}^*) = \gamma^* = \max_{\mathbf{x} \in \mathscr{X}} S(\mathbf{x}) \; . \tag{6}$$

This setting includes many types of optimization problems: discrete (combinatorial), continuous, mixed, and constrained problems. Moreover, if one is interested in minimizing rather than maximizing S, one can simply maximize $-S$.

Now associate with the above problem the estimation of the probability $\ell = \mathbb{P}(S(\mathbf{X}) \geqslant \gamma)$, where $\mathbf{X}$ has some probability density $f(\mathbf{x}; \mathbf{u})$ on $\mathscr{X}$ (for example corresponding to the uniform distribution on $\mathscr{X}$) and γ is some level. If γ is chosen close to the unknown γ^*, then ℓ is, typically, a rare-event probability, and the CE approach of Section 1 can be used to find an importance sampling distribution close to the theoretically optimal importance sampling density, which concentrates all its mass on the point $\mathbf{x}^*$. Sampling from such a distribution thus produces optimal or near-optimal states. A main difference with the CE method for rare-event simulation is that in the optimization setting the final level $\gamma = \gamma^*$ is not known in advance. The CE method for optimization produces a sequence of levels $\{\widehat{\gamma}_t\}$ and reference parameters $\{\widehat{\mathbf{v}}_t\}$ such that the former tends to the optimal γ^* and the latter to the optimal reference vector $\mathbf{v}^*$ corresponding to the point mass at $\mathbf{x}^*$; see, e.g., [46, page 251].

Algorithm 2.1 (CE Algorithm for Optimization)

1. *Choose an initial parameter vector* $\widehat{\mathbf{v}}_0$. *Let* $N^{\mathrm{e}} = \lceil \varrho N \rceil$. *Set* $t = 1$ *(level counter).*

2. *Generate a sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *from the probability density* $f(\cdot; \widehat{\mathbf{v}}_{t-1})$. *Calculate the performances* $S(\mathbf{X}_i)$ *for all* i, *and order them from smallest to largest,* $S_{(1)} \leqslant \ldots \leqslant S_{(N)}$. *Let* $\widehat{\gamma}_t$ *be the sample* $(1-\varrho)$*-quantile of performances; that is,* $\widehat{\gamma}_t = S_{(N-N^{\mathrm{e}}+1)}$.

3. *Use the* **same** *sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *and solve the stochastic program*

$$\max_{\mathbf{v}} N^{-1} \sum_{k=1}^{N} I_{\{S(X_k) \geqslant \widehat{\gamma}_t\}} \ln f(\mathbf{X}_k; \mathbf{v}) \; . \tag{7}$$

   Denote the solution by $\widehat{\mathbf{v}}_t$.

4. *If the stopping criterion is met, stop; otherwise, set* $t = t + 1$, *and return to Step 2.*

To run the algorithm, one needs to provide the class of sampling probability densities, the initial vector $\widehat{\mathbf{v}}_0$, the sample size N, the rarity parameter ϱ, and the stopping criterion. Any CE algorithm for optimization involves thus the following two main iterative phases:

1. **Generate** a random sample of objects in the search space $\mathscr{X}$ (trajectories, vectors, etc.) according to a specified probability distribution.

2. **Update** the parameters of that distribution, based on the N^{e} best performing samples (the so-called *elite samples*), using cross-entropy minimization.

Apart from the fact that Step 3 in Algorithm 1.1 is missing in Algorithm 2.1, another main difference between the two algorithms is that the *likelihood ratio* term $f(\mathbf{X}_k; \mathbf{u})/f(\mathbf{X}_k; \widehat{\mathbf{v}}_{t-1})$ in (5) is missing in (7).

Often a smoothed updating rule is used, in which the parameter vector $\widehat{\mathbf{v}}_t$ is taken as

$$\widehat{\mathbf{v}}_t = \alpha \, \widetilde{\mathbf{v}}_t + (1 - \alpha) \, \widehat{\mathbf{v}}_{t-1}, \tag{8}$$

where $\widetilde{\mathbf{v}}_t$ is the solution to (7) and $0 \leqslant \alpha \leqslant 1$ is a smoothing parameter. Many other modifications can be found in [27, 45, 46] and in the list of references. When there are two or more optimal solutions the CE algorithm typically "fluctuates" between the solutions before focusing in on one of the solutions. The effect that smoothing has on convergence is discussed in detail in [13]. In particular, it is shown that with appropriate smoothing the CE method converges and finds the optimal solution with probability arbitrarily close to 1. Necessary conditions and sufficient conditions under which the optimal solution is generated eventually with probability 1 are also given. Other convergence results,

including a proof of convergence along the lines of convergence for simulated annealing can be found in [33].

2.1 Combinatorial Optimization

When the state space $\mathscr{X}$ is finite, the optimization problem (6) is often referred to as a *discrete* or *combinatorial optimization* problem. For example, $\mathscr{X}$ could be the space of combinatorial objects such as binary vectors, trees, paths through graphs, etc. To apply the CE method, one needs to specify first a convenient parameterized random mechanism to generate objects in $\mathscr{X}$. For example, when $\mathscr{X}$ is the set of binary vectors of length n, an easy generation mechanism is to draw each component independently from a Bernoulli distribution; that is, $\mathbf{X} = (X_1, \ldots, X_n) \sim \mathsf{Ber}(\mathbf{p})$, where $\mathbf{p} = (p_1, \ldots, p_n)$. Given an elite sample set $\mathscr{E}$, the updating formula is then [15, page 56]

$$\widehat{p}_i = \frac{\sum_{\mathbf{X} \in \mathscr{E}} X_i}{N^{\mathrm{e}}}, \quad i = 1, \ldots, n \,. \tag{9}$$

That is, the updated success probability for the i-th component is mean of the i-th components of the vectors in the elite set.

A possible stopping rule for combinatorial optimization problems is to stop when the overall best objective value does not change over a number of iterations. Alternatively, one could stop when the sampling distribution has "degenerated" enough. In particular, in the Bernoulli case (9) one could stop when all $\{p_i\}$ are less than some distance ε away from either 0 or 1.

2.2 Continuous Optimization

It is also possible to apply the CE algorithm to continuous optimization problem; in particular, when $\mathscr{X} = \mathbb{R}^n$. The sampling distribution on $\mathbb{R}^n$ can be quite arbitrary, and does not need to be related to the function that is being optimized. However, the generation of a random vector $\mathbf{X} = (X_1, \ldots, X_n) \in \mathbb{R}^n$ is usually established by drawing the coordinates independently from some 2-parameter distribution. In most applications a normal (Gaussian) distribution is employed for each component. Thus, the sampling distribution for $\mathbf{X}$ is characterized by a vector of means $\boldsymbol{\mu}$ and a vector of standard deviations $\boldsymbol{\sigma}$. At each iteration of the CE algorithm these parameter vectors are updated simply as the vectors of sample means and sample standard deviations of the elements in the elite set; see, for example, [27]. During the course of the algorithm, the sequence of mean vectors ideally tends to the maximizer $\mathbf{x}^*$, while the vector of standard deviations tend to the zero vector. In short, one should obtain a degenerated probability density with all mass concentrated in the vicinity of the point $\mathbf{x}^*$. A possible stopping criterion is to stop when all standard deviations are smaller than some ε.

2.3 Constrained Optimization

Constrained optimization problems can be put in the framework (6) by taking $\mathscr{X}$ a (non-linear) region defined by some system of inequalities:

$$G_i(\mathbf{x}) \leqslant 0, \quad i = 1, \ldots, L \; . \tag{10}$$

To solve the program (6) with constraints (10), two approaches can be adopted. The first approach uses *acceptance–rejection*: generate a random vector $\mathbf{X}$ from, for example, a multivariate normal distribution with independent components, and accept or reject it depending on whether the sample falls in $\mathscr{X}$ or not. Alternatively, one could sample directly from a truncated distribution (for example, a truncated normal distribution) or combine such a method with acceptance-rejection. Once a fixed number of such vectors has been accepted, the parameters of the normal distribution can be updated in exactly the same way as in the unconstrained case — simply via the sample mean and standard deviation of the elite samples. A drawback of this method is that a large number of samples could be rejected before a feasible sample is found.

The second approach is the *penalty approach*. Here the idea is to modify the objective function as follows:

$$\widetilde{S}(\mathbf{x}) = S(\mathbf{x}) + \sum_{i=1}^{L} P_i(\mathbf{x}) \; , \tag{11}$$

where the $\{P_i\}$ are *penalty functions*. Specifically, the i-th penalty function P_i (corresponding to the i-th constraint) is defined as

$$P_i(\mathbf{x}) = H_i \; \max(G_i(\mathbf{x}), 0) \tag{12}$$

and $H_i > 0$ measures the importance (cost) of the i-th penalty.

Thus, by reducing the constrained problem ((6) and (10)) to an unconstrained one ((6) with $\widetilde{S}$ instead of S), one can again apply Algorithm 2.1. Further details on constrained multi-extremal optimization with the CE method may be found in [27].

2.4 Noisy Optimization

Noisy (or stochastic) optimization problems — in which the objective function is corrupted with noise — arise in many contexts, for example, in stochastic scheduling and stochastic shortest/longest path problems, and simulation-based optimization [47]. The CE method can be easily modified to deal with noisy optimization problems. Consider the maximization problem (6) and assume that the performance function is noisy. In particular, suppose that $S(\mathbf{x}) = \mathbb{E}\widehat{S}(\mathbf{x})$ is *not* available, but that a sample value $\widehat{S}(\mathbf{x})$ (unbiased estimate of $\mathbb{E}\widehat{S}(\mathbf{x})$) is available, for example via simulation. The principal modification of the Algorithm 2.1 is to replace $S(\mathbf{x})$ by $\widehat{S}(\mathbf{x})$. In addition, one may need to increase the sample size in order to reduce the effect of the noise. Although various applications indicate the usefulness of the CE approach for noisy optimization (see, for example, [1, 25, 26, 35], little is still known regarding theoretical convergence

results in the noisy case — Spall [50, Section 2.4] discusses various divergence results for general types of stochastic methods. A possible stopping criterion is to stop when the sampling distribution has degenerated enough. Another possibility is to stop the stochastic process when $\{\widehat{\gamma}_t\}$ has reached stationarity; see for example [45, page 207].

References

[1] G. Alon, D. P. Kroese, T. Raviv, and R. Y. Rubinstein. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, 134:137–151, 2005.

[2] S. Asmussen, D. P. Kroese, and R. Y. Rubinstein. Heavy tails, importance sampling and cross-entropy. *Stochastic Models*, 21(1):57–76, 2005.

[3] I. Bendavid and B. Golany. Setting gates for activities in the stochastic project scheduling problem through the cross entropy methodology. *Annals of Operations Research*, 2009. To appear.

[4] Z. I. Botev and D. P. Kroese. Global likelihood optimization via the cross-entropy method, with an application to mixture models. In R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, editors, *Proceedings of the 2004 Winter Simulation Conference*, pages 529–535, Washington, DC, December 2004.

[5] A. Boubezoula, S. Paris, and M. Ouladsinea. Application of the cross entropy method to the GLVQ algorithm. *Pattern Recognition*, 41(10):3173–3178, 2008.

[6] M. Caserta and M. Cabo-Nodar. A cross entropy based algorithm for reliability problems. *Journal of Heuristics*, pages 1381–1231, 2007.

[7] M. Caserta and M. Cabo-Nodar. A cross-entropy based algorithm for combinatorial optimization problems. *European Journal of Operations Research*, 2008.

[8] M. Caserta and E. Quiñonez-Ricob. A cross entropy-Lagrangean hybrid algorithm for the multi-item capacitated lot-sizing problem with setup times. *Computers & Operations Research*, 36(2):530–548, 2009.

[9] J. C. C. Chan and D. P. Kroese. Randomized methods for solving the winner determination problem in combinatorial auctions. In *Proceedings of the 2008 Winter Simulation Conference*, 2008.

[10] J. C. C. Chan and D. P. Kroese. Rare-event probability estimation with conditional Monte Carlo. *Annals of Operations Research*, 2009. To appear.

[11] K. Chepuri and T. Homem-de-Mello. Solving the vehicle routing problem with stochastic demands using the cross entropy method. *Annals of Operations Research*, 134(1):153–181, 2005.

[12] I. Cohen, B. Golany, and A. Shtub. Managing stochastic finite capacity multi-project systems through the cross-entropy method. *Annals of Operations Research*, 134(1):183–199, 2005.

[13] A. Costa, J. Owen, and D. P. Kroese. Convergence properties of the cross-entropy method for discrete optimization. *Operations Research Letters*, 35(5):573–580, 2007.

[14] P. T. de Boer. *Analysis and Efficient Simulation of Queueing Models of Telecommunication Systems*. PhD thesis, University of Twente, 2000.

[15] P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

[16] P. T. de Boer, D. P. Kroese, and R. Y. Rubinstein. A fast cross-entropy method for estimating buffer overflows in queueing networks. *Management Science*, 50(7):883–895, 2004.

[17] A. Eshragh, J. A. Filar, and M. Haythorpe. A hybrid simulation-optimization algorithm for the Hamiltonian cycle problem. *Annals of Operations Research*, 2009. To appear.

[18] G. E. Evans. *Parallel and Sequential Monte Carlo Methods with Applications*. PhD thesis, The University of Queensland, Brisbane, 2009.

[19] J. M. Keith G. E. Evans and D. P. Kroese. Parallel cross-entropy optimization. In *Proceedings of the 2007 Winter Simulation Conference*, pages 2196–2202, Washington, 2007.

[20] B. E. Helvik and O. Wittner. Using the cross-entropy method to guide/govern mobile agent's path finding in networks. In S. Pierre and R. Glitho, editors, *Mobile Agents for Telecommunication Applications: Third International Workshop, MATA 2001, Montreal*, pages 255–268, New York, 2001. Springer-Verlag.

[21] T. Homem-de-Mello. A study on the cross-entropy method for rare event probability estimation. *INFORMS Journal on Computing*, 19(3):381–394, 2007.

[22] K-P. Hui, N. Bean, M. Kraetzl, and D.P. Kroese. The cross-entropy method for network reliability estimation. *Annals of Operations Research*, 134:101–118, 2005.

[23] E. Ianovsky and J. Kreimer. An optimal routing policy for unmanned aerial vehicles (analytical and cross-entropy simulation approach). *Annals of Operations Research*, 2009. To appear.

[24] J. Keith and D. P. Kroese. Sequence alignment by rare event simulation. In *Proceedings of the 2002 Winter Simulation Conference*, pages 320–327, San Diego, 2002.

[25] D. P. Kroese and K-P. Hui. In: *Computational Intelligence in Reliability Engineering*, chapter 3: Applications of the Cross-Entropy Method in Reliability. Springer-Verlag, New York, 2006.

[26] D. P. Kroese, S. Nariai, and K-P. Hui. Network reliability optimization via the cross-entropy method. *IEEE Transactions on Reliability*, 56(2):275–287, 2007.

[27] D. P. Kroese, S. Porotsky, and R. Y. Rubinstein. The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, 2006.

[28] D. P. Kroese and R. Y. Rubinstein. The transform likelihood ratio method for rare event simulation with heavy tails. *Queueing Systems*, 46:317–351, 2004.

[29] D. P. Kroese, R. Y. Rubinstein, and T. Taimre. Application of the cross-entropy method to clustering and vector quantization. *Journal of Global Optimization*, 37:137–157, 2007.

[30] Z. Liu, A. Doucet, and S. S. Singh. The cross-entropy method for blind multiuser detection. In *IEEE International Symposium on Information Theory*, Chicago, 2004. Piscataway.

[31] A. Lörincza, Z. Palotaia, and G. Szirtesb. Spike-based cross-entropy method for reconstruction. *Neurocomputing*, 71(16–18):3635–3639, 2008.

[32] S. Mannor, R. Y. Rubinstein, and Y. Gat. The cross-entropy method for fast policy search. In *The 20th International Conference on Machine Learning (ICML-2003)*, Washington, DC, 2003.

[33] L. Margolin. On the convergence of the cross-entropy method. *Annals of Operations Research*, 134(1):201–214, 2005.

[34] I. Menache, S. Mannor, and N. Shimkin. Basis function adaption in temporal difference reinforcement learning. *Annals of Operations Research*, 134(1):215–238, 2005.

[35] S. Nariai. *Cross-Entropy Methods in Telecommunication Systems*. PhD thesis, The University of Queensland, Brisbane, 2009.

[36] S. Nariai and D. P. Kroese. On the design of multi-type networks via the cross-entropy method. In *Proceedings of the Fifth International Workshop on the Design of Reliable Communication Networks (DRCN)*, pages 109–114, 2005.

[37] S. Nariai, D. P. Kroese, and K-P. Hui. Designing an optimal network using the cross-entropy method. In *Intelligent Data Engineering and Automated Learning*, Lecture Notes in Computer Science, pages 228–233, New York, 2005. Springer-Verlag.

[38] V. Pihur, S. Datta, and S. Datta. Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. *Bioinformatics*, 23(13):1607–1615, 2007.

[39] A. Ridder. Importance sampling simulations of Markovian reliability systems using cross-entropy. *Annals of Operations Research*, 134(1):119–136, 2005.

[40] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99:89–112, 1997.

[41] R. Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 2:127–190, 1999.

[42] R. Y. Rubinstein. Combinatorial optimization, cross-entropy, ants and rare events. In S. Uryasev and P. M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, pages 304–358, Dordrecht, 2001. Kluwer.

[43] R. Y. Rubinstein. Combinatorial optimization via cross-entropy. In S. Gass and C. Harris, editors, *Encyclopedia of Operations Research and Management Sciences*, pages 102–106. Kluwer, 2001.

[44] R. Y. Rubinstein. The cross-entropy method and rare-events for maximal cut and bipartition problems. *ACM Transactions on Modelling and Computer Simulation*, 12(1):27–53, 2002.

[45] 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.

[46] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, second edition, 2007.

[47] R. Y. Rubinstein and B. Melamed. *Modern Simulation and Modeling*. John Wiley & Sons, New York, 1998.

[48] A. Sani. *Stochastic Modelling and Intervention of the Spread of HIV/AIDS*. PhD thesis, The University of Queensland, Brisbane, 2009.

[49] A. Sani and D. P. Kroese. Controling the number of HIV infectives in a mobile population. *Mathematical Biosciences*, 213:103–112, 2008.

[50] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, New York, 2003.

[51] T. Taimre. *Advances in Cross-Entropy Methods*. PhD thesis, The University of Queensland, Brisbane, 2009.

[52] A. Unveren and A. Acan. Multi-objective optimization with cross entropy method: Stochastic learning with clustered pareto fronts. In *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3065–3071, 2007.

[53] J. Wang, X. Gao, J. Shi, and Z. Li. Double unmanned aerial vehicle's path planning for scout via cross-entropy method. In *8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, volume 2, pages 632–635, 2007.

[54] Y. Wu and C. Fyfe. Topology perserving mappings using cross entropy adaptation. In *7th WSEAS international conference on artificial intelligence, knowledge engineering and data bases*, Cambridge, 2008.