The Cross-Entropy Method

compiled by D.Gueorguiev 3/20/25

# Introductory Notes

The travelling salesman problem (TSP), the quadratic assignment problem (QAP), and the max-cut problem (MCP) are combinatorial optimization problems where the problem studied is completely known and static. In contrast, the buffer allocation problem (BAP) is a noisy estimation problem where the objective function needs to be estimated since it is unknown. Discrete event simulation is one method for estimating an unknown objective function.

The Cross-Entropy method (CE) is a general and efficient method for solving problems of the discussed types above.

Moreover, CE is also valuable for *rare event-simulation*, where very small probabilities need to be accurately estimated. The CE method was motivated by an adaptive algorithm for estimating probabilities of rare events in complex stochastic networks which involve variance minimization (Rubinstein, 1997). It was soon realized that a simple cross-entropy modification of the original adaptive algorithm could be used not only for estimating probabilities of rare events but for solving difficult combinatorial optimization problems (COP) as well. This is done by translating the deterministic optimization problem into a related stochastic optimization one and then using rare event simulation techniques to solve it.

The CE method involves an iterative procedure where each iteration can be broken down into 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 better sample in the next iteration

The significance of the CE method is that it defines a precise mathematical framework for deriving fast and in some sense optimal updating/learning rules based on simulation theory.

Many COPs can be formulated as optimization problems concerning a weighted graph. As mentioned in previous paragraph, in CE a deterministic optimization problem is translated into an associated *stochastic optimization problem*. Depending on the particular problem, we introduce randomness in either (a) the nodes or (b) the edges of the graph. We are dealing with *stochastic node networks* (SNN) in the former case and *stochastic edge networks* (SEN) in the latter. Examples of SNN problems are the maximal cut (max-cut) problem, the buffer allocaton problem and clustering problems. Examples of SEN problems are the travelling salesman problem, the quadratic assignment problem, the clique problem, and optimal policy search in Markovian Decision Problems (MDP).

The CE method can be successfully applied to both deterministic and stochastic COPs. In the latter the objective function itself is random or needs to be estimated via simulation. Stochastic COPs typically occur in stochastic scheduling, flow control and routing of data networks and in various simulation-based optimization problems such as the BAP.

Estimation of the probability of rare events is essential for guaranteeing adequate performance of engineering systems. For example, consider a telecommunication system that accepts calls from many customers. Under normal operating conditions each client may be rejected with a very small probability. Naively, to estimate this small probability, we would need to simulate the system under normal operating conditions for a long time. A better way to estimate this probability is to use *importance sampling* (IS), which is a well-known variance reduction technique in which the system is simulated under a different set of parameters – or, more generally, a different probability distribution – so as to make the occurrence of the rare event more likely. A major drawback of the IS technique is that the optimal reference (also called *tilting*) parameters to be used in IS are usually very difficult to obtain. The advantage of the CE method is that it provides a simple adaptive procedure for estimating the optimal reference parameters. Moreover, the CE method also enjoys asymptotic convergence properties. For example, it is shown that for *static models* under mild regularity conditions the CE method terminates with probability 1 in a finite number of iterations and delivers a consistent and asymptotically normal estimator for optimal reference parameters. The CE method has been applied successfully to the estimation of rare event probabilities in dynamic models, in particular queueing models involving *light* and *heavy* tail input distributions. In addition to rare simulation and combinatorial optimization, the CE method can be efficiently applied to continuous multi-extrema optimization (see [5]).

# Examples

## A Rare Event Simulation Example

# References

[1] [A Tutorial On The Cross-Entropy Method, Pieter-Tjerk de Boer et al, MIT, 2003](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/A_Tutorial_on_the_Cross-Entropy_Method_deBoer_2003.pdf)

[2] [Learning Tetris Using the Noisy Cross-Entropy Method, Istvan Szita, Andras Loerincz, 2006](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Learning_Tetris_Using_the_Noisy_Cross-Entropy_Method_Szita_2006.pdf)

[3] [Approximate Dynamic Programming Finally Performs Well In The Game of Tetris, Victor Gabillon et al, INRIA, 2013](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/NIPS-2013-approximate-dynamic-programming-finally-performs-well-in-the-game-of-tetris-Paper.pdf)

[4] <https://en.wikipedia.org/wiki/Cross-entropy_method>

[5] [The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning, RY Rubinstein, DP Kroese, 2004](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/books/The_Cross_Entropy_Method_A_Unified_Approach_Rubinstein_Kroese_2004.pdf)

[6] [Application of the Cross-Entropy Method to the Buffer Allocation Problem in a Simulation-Based Environment, G. Allon et al, 2005](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/stochastic_optimization/Application_of_the_Cross-Entropy_Method_to_the_Buffer_Allocation_Problem_in_a_Simulation-Based_Environment_Allon_2005.pdf)

[7] [Buffer Allocation Problem in production flow lines: A new Benders-decomposition-based exact solution approach, M. Zhang et al, 2022](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/stochastic_optimization/Buffer_Allocation_Problem_in_production_flow_lines_new_Benders-decomposition-based_exact_solution_approach_Zhang_2022.pdf)

[8] [Introduction to Rare Events Simulation, John F. Shortle, Pierre L'Eccuyer, Draft, 2011](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/stochastic_optimization/Introduction_to_Rare-Event_Simulation.pdf)

[9] [Methods to optimize rare-event Monte Carlo reliability simulations for Large Hadron Collider Protection Systems, Milosz Blaszkiewicz, Thesis, 2022](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/stochastic_optimization/Methods_to_optimize_rare-event_Monte_Carlo_reliability_simulations_for_Large_Hadron_Collider_Protection_Systems_CERN-Thesis-2022.pdf)

# Appendix

## The Buffer Allocation Problem

The BAP finds the optimal buffer capacity on each stage in a flow line i.e. the minimum buffer capacity able to guarantee the target throughput.

Let us consider a flow as depicted in the Figure below:

A diagram of a block diagram

Description automatically generated

Figure 1: flow line under consideration in the BAP

There are machines, depicted with circles, and inter-machine buffers of finite capacity . All parts are processed by all machines of the line, ordered according to the arrival sequence. Parts are always available in front of the first machine, and parts can immediately leave the system after being processed by the last machine. The inter-machine buffer has finite capacity . A full buffer causes the *blocking* of the upstream machine, and an empty buffer causes the *starvation* of the downstream machine. The processing times of each machine can be generated as independent and identically distributed variables from a general distribution or calculated as the sum of the two variables corresponding to the processing and the repair time, if the machine is considered unreliable. The term *processing time* will represent the processing time of reliable machines, and the processing time without including the repair time of unreliable machines.

**Notation**

index , set and number of system stages

set of all the stages but the last one

lower and upper bound of the capacity of buffer

index, set and number of parts in the simulation horizon

processing time of part at machine

buffer capacity index

set of all the possible capacities for buffer except the lower bound

big-M and small-M parameters

target throughput

The BAP model is formulated as:

s.t. (1)

(2)

(3)

//TODO: finish the BAP formulation