Randomized Algorithms for Counting, Rare-Events Estimation and Optimization

Reuven Rubinstein

Faculty of Industrial Engineering and
Technion, Israel Institute of Technology,
Haifa , Israel

ierrr01@ie.technion.ac.il

iew3.technion.ac.il:8080/ierrr01.phtml

Abstract

We present a mini-course on the classic randomized algorithm, and the new splitting algorithm, for approximate counting in NP-hard problems, rare-event estimation in static and dynamic (queueing) models, solving quite general NP-hard combinatorial optimization problems, and for uniform sampling on complex regions. Both the classic randomized algorithm and the splitting algorithm are based on the MCMC (Markov chain Monte Carlo) sampler and and they use a sequential sampling plan to decompose a "difficult" problem into a sequence of "easy" ones.

We show that the *original classic randomized algorithms* for approximate counting, presenting a particular case of our splitting algorithms, typically fail. They either do not converge at all or deliver a heavily biased estimator (converge to a local extremum). Exceptions are convex counting problems, like estimating the volume of a convex polytope. As compared to the randomized algorithms, the new splitting algorithms, which are based on simulating simultaneously multiple Markov chains, require very little warm-up time while running the MCMC from iteration to iteration, since the underlying Markov chains are already in steady-state from the beginning. What required is only fine tuning, i.e. keeping the Markov chains in steady-state while moving from iteration to iteration.

We also present a combined version of the splitting and cross-entropy (CE) algorithms and prove the polynomial complexity of a particular version of our algorithm.

We, finally present supportive numerical results with the new splitting algorithm and its combined version, while solving quite general counting problems, like counting the number of satisfiability assignments in a SAT problem, counting the number of feasible colorings in a graph, calculating the permanent, Hamiltonian cycles, 0-1 tables, volume of a polytope, as well as solving integer and combinatorial optimization, like TSP, Knapsack and set covering problems.