Monte Carlo Algorithms for Quantum Systems

Mark Dubynskyi, Raghavendra Guggilam, Andrew Miller, John Kent, Anthony E. Pizzimenti, Michael Jarret-Baume

Mason Experimental Geometry Lab

MEGL Final Symposium — April 26th, 2024

Quantum Monte Carlo Introduction

Our group dives deep into the notion of Quantum Monte Carlo for Quantum Algorithms. Quantum Monte Carlo (QMC) methods are a class of computational algorithms used to solve quantum problems where exact solutions are often not feasible due to their computational complexity.

Metropolis-Hastings

The Metropolis–Hastings algorithm is a Markov chain Monte Carlo method which can obtain a sequence of random samples from a probability distribution when direct sampling is infeasible.

Simulated Annealing

Algorithm 1 Simulated Annealing

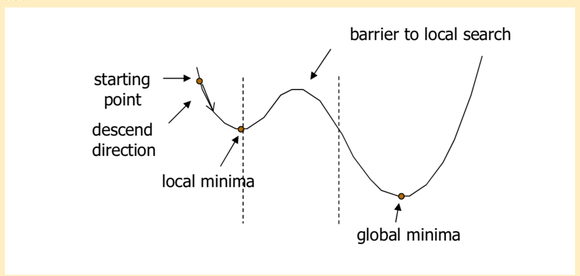
function SimulatedAnnealing(T, k_{max})
  $w \leftarrow$ dist(V)
  $w_{min} \leftarrow w$

  for $k = 0$ **to** $k_{\max} - 1$ **do**
    $T \leftarrow$ temperature $\left(1 - \frac{k+1}{k_{\max}}\right)$
    $u \leftarrow$ ProposeUpdate(w_i)

    $w_i \leftarrow u$ with probability $\min\{1, \text{Cost}(w_i, t)/\text{Cost}(u, t)$
    if Cost(w_i) < Cost(w_{min}) **then**
      $w_{min} \leftarrow w_i$

  return w_{min}

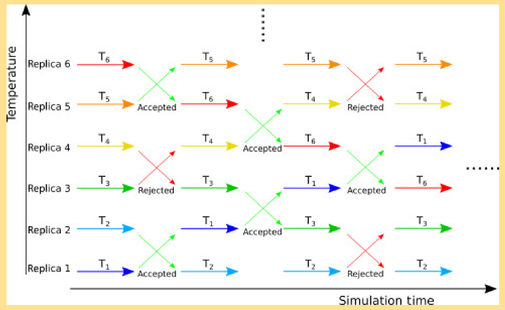
Simulated annealing is a metaheuristic for finding a good approximation of the global optimum of a given function. Simulated annealing mimics the physical process of heating and then slowly cooling a material, moving the system to neighboring states with an acceptance probability based on the energy of current state, the energy of a proposed neighboring state, and a temperature. The temperature cools periodically, reducing the likelihood of moving towards higher energy states, thus settling into a minimum-energy state.



Parallel Tempering

Overview

Parallel tempering is essentially an improved version of simulated annealing. Parallel tempering, (aka replica exchange Monte Carlo), is an advanced Monte Carlo simulation technique used to enhance the sampling efficiency of a system, especially when the system's energy landscape contains many local minima. This method involves running multiple simultaneous simulations (or "replicas") of the same system at different temperatures. Higher temperatures allow the system to overcome energy barriers and escape local minima, exploring the energy landscape more broadly.



Pseudocode

We formalized the parallel tempering algorithm by creating pseudocode based on existing work. Let T be a finite subset of the real numbers (representing the "energies" of the replicas), and let h be a stopping condition.

Algorithm 2 Parallel Tempering

function ParallelTempering(T, h)
  $W \leftarrow$ empty array of length $|T|$
  $w_i \leftarrow$ dist(V) for $0 \leq i < |T|$
  $w_{min} \leftarrow w_i$
  while !h(W) **do**
    for $i \in \{0, \ldots, |T| - 1\}$ **do**
      $u \leftarrow$ ProposeUpdate(w_i)
      $w_i \leftarrow u$ with probability $\min\{1, \text{Cost}(w_i, t)/$
      if Cost(w_i) < Cost(w_{min}) **then**
        $w_{min} \leftarrow w_i$
    SORT(W)
  return w_{min}

Logic Synthesis

One of the problems to which we apply parallel tempering is the Logic Synthesis problem [2]. This problem seeks to reduce the number of simple gates to recreate higher-level functions. In particular, we are trying to improve on the state of the art for synthesis of majority-n gates out of majority-3 gates for odd values of n. These circuits are represented using ternary graphs. The energy (or cost) of a state (one logic circuit) is determined by the number of inputs that result in an incorrect majority. We say that $E(N) = 0$ if, and only if,

$$f(x) = N(x) \quad \forall x \in \{0,1\}^n.$$

If $N(x)$ is the result of running an input through a network and f is the boolean value function to imitate, then the cost of a network is

$$E(N) = \sum_{x \in \{0,1\}^n} f(x) \oplus N(x).$$

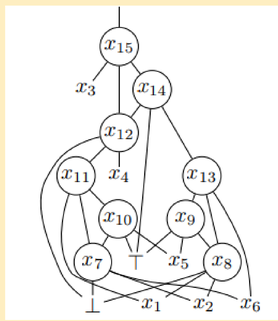


Figure: [4, Fig. 2] Ternary graph representing synthesis using majority-3 gates

Conclusions/Future Work

Code and Software Development. We are in the process of developing libraries for executing parallel tempering in an efficient manner over a given state space.

Algorithm Classification and Analysis. We are currently discussing defining a new class or flavor of algorithms which could include simulated annealing, parallel tempering, and go-with-the-winners, among others. We are currently analyzing these algorithms as graph search algorithms where the graph being searched is the state space.

Development of New Parallel Tempering Algorithms. Developing algorithms that dynamically adjust the temperatures of the replicas during the simulation can potentially improve efficiency. This involves creating adaptive methods that can determine the optimal temperature distribution based on the current state of the system. Exploring different rules for exchanging replicas, beyond the traditional Metropolis criterion, could lead to more effective sampling.

Acknowledgments

We would like to give special thanks to Anton Lukyanenko, Swan Klein, and all those who run and support MEGL.

References

[1] P. Grassberger, "Go with the winners: a general Monte Carlo strategy," Computer Physics Communications, vol. 147, no. 1, pp. 64–70, Aug. 2002, doi: 10.1016/S0010-4655(02)00205-9.

[2] T. Häner, D. S. Steiger, and H. G. Katzgraber, "Parallel Tempering for Logic Synthesis." arXiv, Nov. 21, 2023. Accessed: Jan. 26, 2024. [Online]. Available: http://arxiv.org/abs/2311.12394

[3] M. Jarret and B. Lackey, "Substochastic Monte Carlo Algorithms." arXiv, Apr. 28, 2017. doi: 10.48550/arXiv.1704.09014.

[4] E. Testa, M. Soeken, L. G. Amaru, W. Haaswijk, and G. De Micheli, "Mapping Monotone Boolean Functions into Majority," IEEE Trans. Comput., vol. 68, no. 5, pp. 791–797, May 2019, doi: 10.1109/TC.2018.2881245.

[5] J.-S. Wang and R. H. Swendsen, "Replica Monte Carlo Simulation (Revisited)," Prog. Theor. Phys. Suppl., vol. 157, pp. 317–323, 2005, doi: 10.1143/PTPS.157.317.