Monte Carlo for Quantum Systems

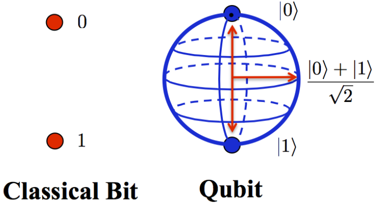
Mark Dubynskyi, Raghu Guggilam, Andy Miller
Michael Jarret-Baume, John Kent, Anthony E. Pizzimenti

George Mason University, MEGL

April 26th, 2024

Quantum Computing - Why Do We Care?

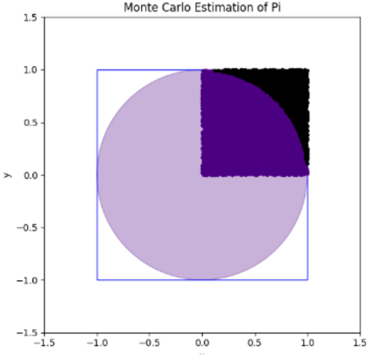
Type of computing that uses quantum-mechanical phenomena (superposition and entanglement) to perform operations on data.



In order to find out when quantum algorithms can provide a speedup in optimization problems, we need to understand the best classical algorithms.

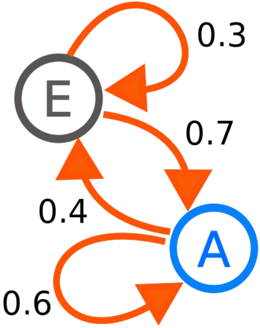
Last Time on MEGL

Monte Carlo



Statistical technique that utilizes random sampling to solve problems and compute results.

Markov Chains

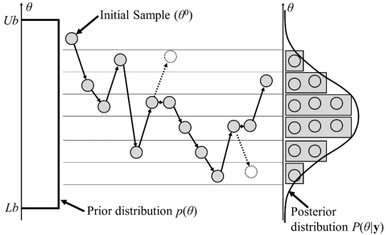


Stochastic model for event sequences where the probability of an event depends only on the previous state.

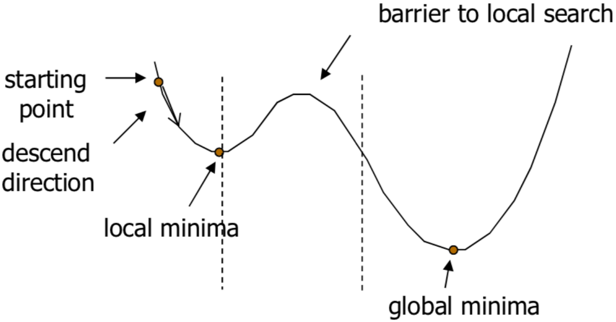
Metropolis-Hastings

- A stochastic sampling technique used in statistical computing to generate a sequence of samples from a probability distribution where direct sampling is difficult.

- Constructs a Markov chain that has the desired distribution as its equilibrium. The samples generated by this chain can be used to approximate the target distribution after a sufficient number of steps.



Simulated Annealing



Optimization technique that uses random variations and gradual cooling to find a global minimum of a function.

Simulated Annealing

Algorithm 1 Simulated Annealing

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

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

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

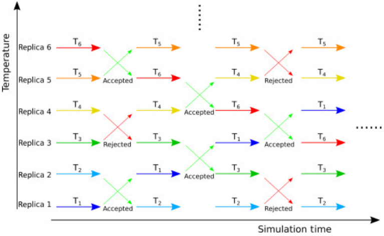
  return w_{min}

Parallel Tempering

- Robust method used in computational physics, chemistry, and biology but never formalized in a context-independent way [2, 4].

- Improves the sampling efficiency from complex probability distributions by creating multiple replicas of the system which all use Metropolis-Hastings.

- Useful in systems where the probability landscape has multiple local minima, which can make sampling inefficient for standard Monte Carlo methods.



Our Advance: Pseudo-code

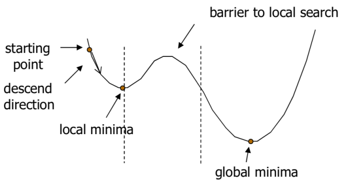
Let T be a finite subset of the real numbers (representing the “energies” of the replicas), and let h be a stopping condition.

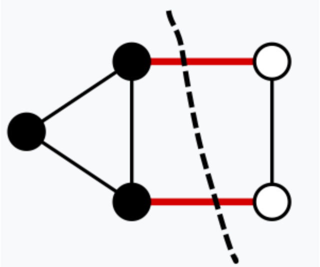
Algorithm 2 Parallel Tempering

function $\textsc{ParallelTempering}(T, h)$
    $W \leftarrow$ 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**
            $u \leftarrow \textsc{ProposeUpdate}(w_i)$
            $w_i \leftarrow u$ with probability $\min\{1, \textsc{Cost}(w_i, t)/\textsc{Cost}(u, t)\}$
            if $\textsc{Cost}(w_i) < \textsc{Cost}(w_{min})$ **then**
                $w_{min} \leftarrow w_i$
        $\mathrm{SORT}(W)$
    return w_{min}

Optimization →Graph Search

We began analyzing parallel tempering in terms of searching a graph as opposed to optimization.





- A barrier to local search where a temperature doesn't allow a walker to cross is equivalent to a cut in the graph.
- For a well-conditioned graph, parallel tempering can work like binary search.

Using temperature to induce a cut

- The probability of accepting α a new state depends on temperature T and the difference between the energy E of the current state and proposed state.
  Let $\alpha = \frac{e^{-\frac{1}{T}(E(x'))}}{e^{-\frac{1}{T}(E(x))}} = e^{-\frac{1}{T}[E(x')-E(x)]}$
  Let $\delta = E(x') - E(x)$

- Doubling T will result in an α that is $\sqrt{\alpha}$ times larger.
  $\alpha_1 = e^{-\frac{1}{T_1}[\delta]}$
  $\alpha_2 = e^{-\frac{1}{T_2}[\delta]}$
  Let $T_2 = 2T_1$
  $\frac{\alpha_2}{\alpha_1} = e^{-\frac{1}{2T_1}[\delta]}$
  $\alpha_2 = (\alpha_1)^{\frac{3}{2}}$

- The acceptance rate α grows exponentially with respect to temperature T.

Using temperature to induce a cut

- Given

$$P_{accept}(T) = \alpha$$

- We can choose a desired acceptance rate using

$$P_{accept}(2^n T) = \alpha^{\frac{3n}{2}}$$

Logic Synthesis

- Crucial process in the design and development of digital circuits.
- The process begins with a high-level description of a digital circuit. This description specifies the behaviour of the circuit in terms of logic functions without detailing the physical implementation [3].

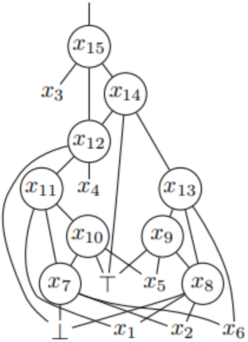


Figure: *Image retrieved from [3].*

Our Work on Logic Synthesis

- We apply parallel tempering to the Logic Synthesis problem [1]. This problem seeks to reduce the number of simple gates to recreate higher-level functions.

- We improve the state of the art for synthesis of majority-n gates.

- These operations are represented as directed ternary trees.

- We say that $E(N) = 0$ if, and only if,

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

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

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

Hoeffding's Inequality

- One potential direction for future work is analyzing Parallel Tempering's runtime using Hoeffding's Inequality.
- Provides an upper bound on the probability of a sequence of random variables deviating from their expected values
- This is important for our logic synthesis problem, since calculating cost can be computationally expensive!

New class of algorithms

Simulated annealing, parallel tempering, and go-with-the-winners all seem to use a common set of actions {stay, walk, jump}:

- stay: when a walker doesn't go to a proposed adjacent state
- walk: when a walker does go to an adjacent state
- jump: when a walker goes to a non-adjacent state

Additional Future Work

- Programming libraries for executing parallel tempering with machine learning libraries that could open up new applications, particularly in optimizing neural network parameters.
- Creating algorithms that dynamically adjust the temperatures of the replicas during the simulation can potentially improve efficiency.
- Developing more rigorous proofs of convergence rates and conditions under which convergence occurs.

References and Acknowledgements

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

[1] Thomas Häner, Damian S. Steiger, and Helmut G. Katzgraber. "Parallel Tempering for Logic Synthesis". en. In: arXiv:2311.12394 (Nov. 2023). arXiv:2311.12394 [cs]. URL: http://arxiv.org/abs/2311.12394.

[2] Robert H. Swendsen and Jian-Sheng Wang. "Replica Monte Carlo Simulation of Spin-Glasses". In: *Physical Review Letters* 57.21 (Nov. 1986), pp. 2607–2609. DOI: 10.1103/PhysRevLett.57.2607.

[3] Eleonora Testa et al. "Mapping Monotone Boolean Functions into Majority". en. In: *IEEE Transactions on Computers* 68.5 (May 2019), pp. 791–797. ISSN: 0018-9340, 1557-9956, 2326-3814. DOI: 10.1109/TC.2018.2881245.

[4] Jian-Sheng Wang and Robert H. Swendsen. "Replica Monte Carlo Simulation (Revisited)". In: *Progress of Theoretical Physics Supplement* 157 (2005). arXiv:cond-mat/0407273, pp. 317–323. ISSN: 0375-9687. DOI: 10.1143/PTPS.157.317.