

Quantum Monte Carlo for Quantum-dot Cellular Automata

Ezra Savard

for Konrad Walus, Electrical and Computer Engineering, University of British Columbia
24 April 2016

A fast solver that can find ground state solutions while providing information about the occupation of states in quantum annealing (QA) systems is very desirable. The path-integral Quantum Monte Carlo (QMC) as described by Martonak et al. is an interesting candidate due to having been modeled on physical quantum annealing [1]. While this implementation of QMC behaved similarly, it did not achieve the same levels of performance as reported by Martonak or Denchev [1][2]. Some interesting results were obtained simulating the final distributions of physical quantum annealing on quantum-dot cellular automata (QCA) circuits, but more work is needed on this front before it could become useful.

1. INTRODUCTION

Quantum annealing is the process at the core of modern quantum annealing computers like the D-Wave 2X. These computers can be used to solve some difficult problems very quickly with applications in areas such as genomics, finance, infrastructure planning, and computer vision [3][4].

Quantum-dot cellular automata is a quantum annealing computing technology that is predicted to offer low power consumption and high switching speeds. QCA could serve as the building blocks for future quantum annealing processors [5][6][7].

While members of our group are currently simulating QCA circuits physically on D-Wave's quantum annealing processor, this work attempts to replicate their simulation results on a classical computer by implementing the path-integral Quantum Monte Carlo (QMC) algorithm. This algorithm is modeled after quantum annealing and based on work by Martonak et al. QMC may have potential in simulating the physical quantum annealing process [1][2].

Being able to accurately simulate quantum annealing devices on classical computers would be valuable in designing future quantum annealing systems. QCADesigner [9], the dominant design tool for QCA, currently simulates circuits by solving for ground states only, rather than a probabilistic distribution of states. Additionally, some approximations used in solving the circuits, like the intercellular Hartree approximation, may produce metastable states instead of the ground state of the system [3][5].

Though well-known heuristic solvers like classical simulated annealing (SA) can quickly find ground

states, they do not provide any information about the probability distribution states.

A probabilistic solver that can overcome meta-stability to find ground states while providing information about the final state distribution would be very valuable. This work explores QMC as a potential candidate.

2. QMC ORIGINS AND PARAMETERS

The Quantum Monte Carlo algorithm is derived from the path-integral formulation of quantum mechanics. The quantum Hamiltonian is the actual cost function for a QCA circuit.

The path-integral approximation maps a d-dimensional quantum Hamiltonian to a (d+1)-dimensional classical Hamiltonian summed over P replicas, known as Trotter slices. The transverse field is replaced with an inter-slice coupling term.

$$H_d = -\epsilon \left(\sum_{i<j} J_{ij} \sigma_z^i \sigma_z^j + \sum_i h_i \sigma_z^i \right) + \Gamma \sum_i \sigma_x^i$$

$$H_{d+1} = -\epsilon \sum_{k=1}^P \left(\sum_{i<j} J_{ij} s_i^k s_j^k + \sum_i h_i s_i^k + J_{\perp} \sum_i s_i^k s_i^{k+1} \right)$$

The inter-slice coupling term, J-perpendicular is

$$J_{\perp} = -\frac{PT}{2} \ln \tanh \left(\frac{\Gamma}{PT\epsilon} \right)$$

The algorithm then uses the classical Hamiltonian as a cost function to minimize the energy of each coupled slice. The slices start out mostly independent and, an increasing J-perpendicular pulls the slices

toward alignment near the end of the process. The schedule used is shown in figure 1 below.

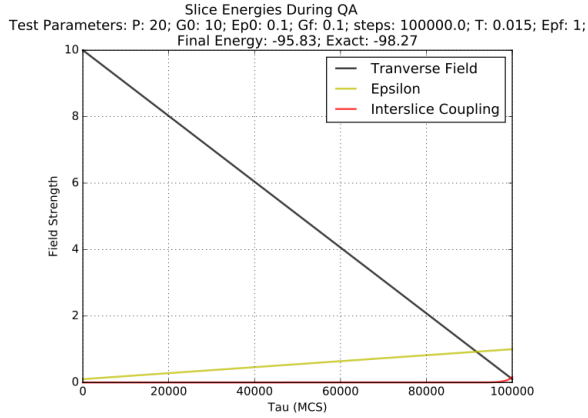


Figure 1. QMC annealing schedule used. A large initial transverse field and small epsilon promote tunneling in the system in the early stages of QMC. Near the end of the process, the inter-slice coupling term grows rapidly, forcing the slices to converge.

QMC can be tuned by adjusting the temperature bath (T), transverse field strength (G), coupling strength pre-factor (Ep) and the number of slices (P). It is recommended to keep the product PT in the range of the average coupling strengths for the problem. Finally, the number of steps per spin is specified and largely determines the length of computation. For a full derivation of the algorithm and details about the parameters, see Martonak’s paper on the topic [1].

3. METHODS

Tests were carried out to validate QMC as a ground state solver and as a simulator for finding the final state distributions.

All comparison testing between SA and QMC was carried out by performing several iterations and averaging the results for each step count. For QMC, the final energy value used in the average is the energy value of the lowest energy Trotter slice at the end of the process. For SA, the value used in the average is the lowest energy value from the last thousand steps.

Every attempt was initialized with a random spin configuration. The pre-annealing used by Martonak et al. was dropped, as this had no noticeable impact on the relative performance of the algorithms.

A total of four problems were used in testing.

Majority Gate

The majority gate is a small (10 spins) QCA logic gate used to select the majority agreement of three inputs. This paper uses a 531-majority gate, shown in figure 2, which has arm lengths of 5, 3 and 1 QCA cells. It was tested with [-1 -1 +1] inputs respectively.

This is considered a worst-case for meta-stability in the circuit.

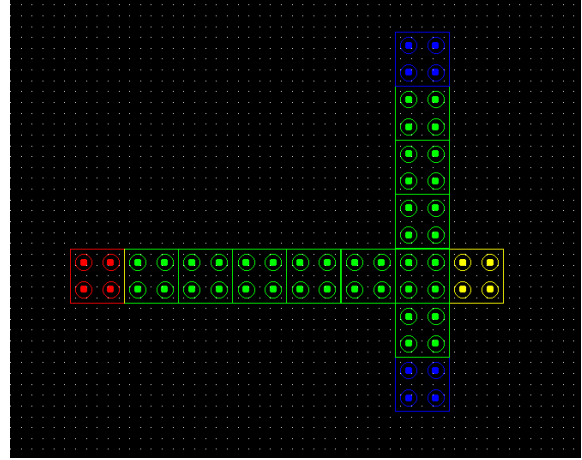


Figure 2. A 531 Majority Gate in QCADesigner. The red and blue coloured input and output cells are not included in this simulation.

XOR Gate

The exclusive-or gate is a larger QCA circuit with 102 spins, this circuit was large enough to pose a challenge in solving while having physical QA results available.

Wires

QCA wires are one-dimensional and have trivial physics. Wires of length 10, 20, 30, and 50 were used. The wires have available solutions and were helpful in examining the computational overhead of the algorithm.

Ising32

A random 1024 spin problem with a known solution from the Spin Glass Server [8], this problem was chosen as neither QMC nor SA would be able to reliably reach the ground state in a reasonable amount of time.

Final distributions were calculated over ten iterations for the XOR gate and Ising32 problem and forty iterations for the majority gate and wires. The total number of slices present in each state were compared to the physical results from simulating circuits on the D-Wave 2X. The physical results were scaled to have the same total count as QMC.

4. QMC AS A SOLVER

QMC is well regarded as a solver and has been shown to outperform SA on problems similar to Ising32. Further, QMC generally exhibits better scaling with problem size and finds lower energy states than classical simulated annealing [1][2]. This work was not able to reproduce these performance

metrics, though the algorithm did generally behave as expected otherwise. In small, sparse problems like QCA circuits, this QMC worked, but SA usually outperformed it; this might be an implementation issue.

Wires

QMC found the ground state of a wire in most of the 40 trials attempted. As shown in the figure below, average residual energies were fairly small, but non-zero, pointing to an occasional failure to find the ground state.

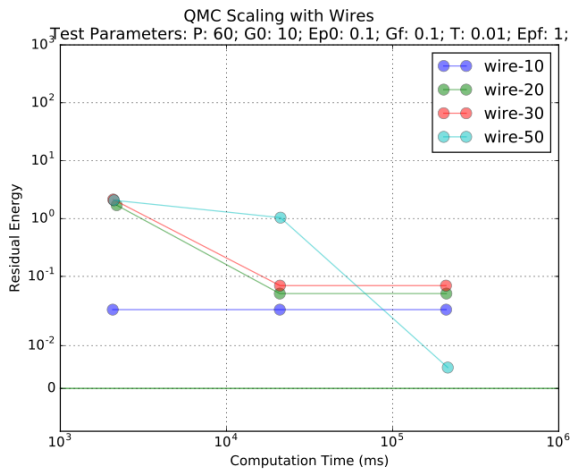


Figure 3. Solving with various step sizes on different length QCA wires. QMC is not always achieving the ground state, but arriving within small residual energies in similar time for different length wires.

Simulated annealing samples the final solution from the last thousand values, while this QMC implementation is only able to draw from P slices, which made this QMC more vulnerable to final step fluctuations.

XOR Gate

As shown in figure 4, QMC was able to approach the ground state solution to this circuit, but SA did so faster. The XOR gate was a useful testing ground for tuning the numerous parameters used by QMC.

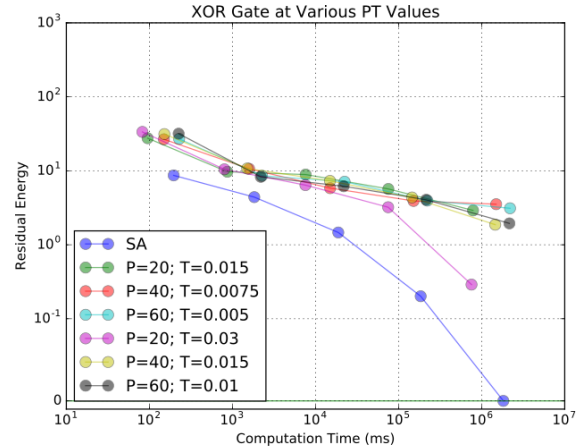


Figure 4. Solving the XOR gate with various parameters show significantly better behaviour from PT=0.6 (pink, yellow, black), particularly with P=20. In this case, classical simulated annealing finds the ground state in all trials on the longer attempts.

Ising32

Figure 5 shows that QMC was outperformed similarly by SA on this problem as with the XOR gate. This problem is similar to those used by Martonak et al. in their 2003 paper in which QMC outperformed SA significantly [1].

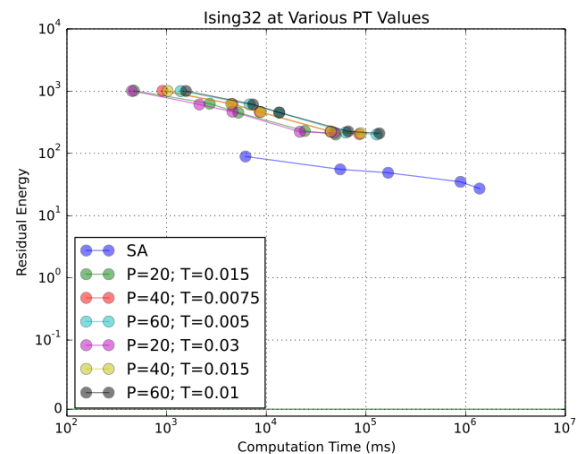


Figure 5. Solving Ising32 with various parameters no significant improvements from parameter selection other than increasing step count on this problem. A larger step count would likely reveal parameter differences similar to the XOR gate.

5. QMC AS A SIMULATOR

In addition to functioning as a solver, QMC can provide information on the probability of final state occupation for the circuits. These probabilities were compared to physical QA results obtained from tests on the D-Wave 2X and were not in very good agreement.

Majority Gate

For the majority gate, figures 6 and 7 show that the final states from QMC have a very similar distribution to the physical D-Wave results.

Operating QMC at a cold temperature ($PT = 0.3$) resulted in a distribution that matched the distribution from physical QA well. Warm temperature ($PT = 0.6$) attempts gave inferior results for final state probability distribution. In both cases, operating at 100 steps was insufficient to replicate the correct distribution, while 1000 steps was sufficient for cold simulation runs.

At first glance this implies that QMC can be used to simulate a quantum annealing process, given the right parameters, but this is potentially misleading. It is likely that sufficiently long trials will result in a fully populated of the ground state regardless of the correct physical distribution.

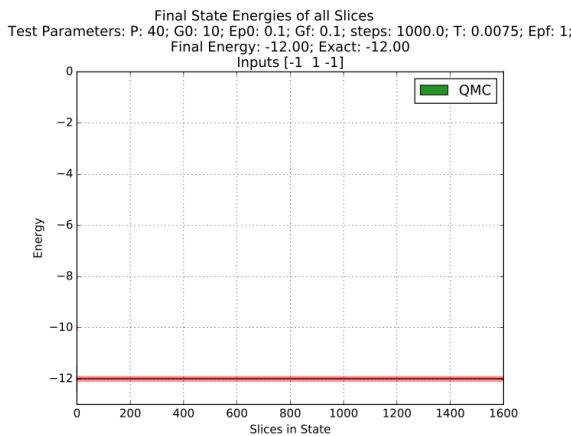


Figure 6. Final state distributions for 531 majority gate with $PT=0.3$. The physical QA distribution is shown in red and is almost entirely the ground state value.

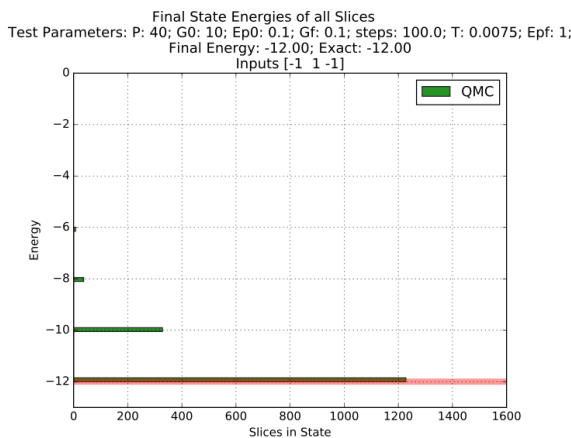


Figure 7. Final state distributions for 531 majority gate with $PT=0.6$. The physical QA distribution is shown in red and is almost entirely the ground state value.

XOR Gate

The final state distributions for the XOR gate, shown in figures 8 and 9, were not simply a fully populated ground state. The QMC simulated results differed substantially from the physical quantum annealing results. Unlike with the majority gate, a hot ($PT=0.6$) simulation resulted in a better solution for the XOR gate.

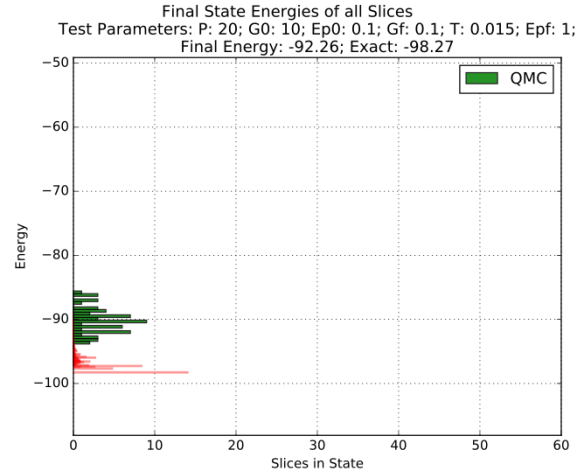


Figure 8. Final state distributions for the XOR gate with $PT=0.3$. The physical QA distribution is shown in red.

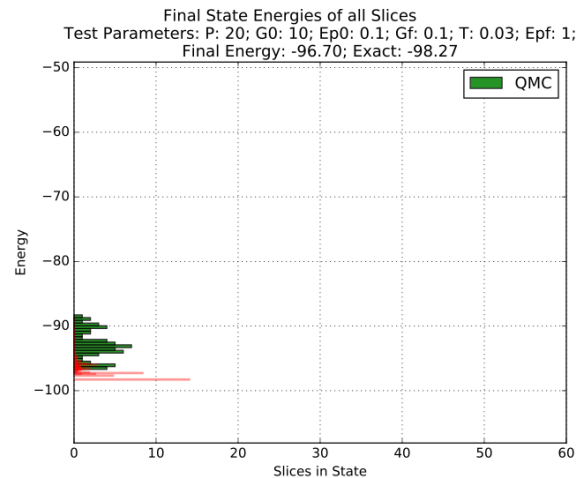


Figure 9. Final state distributions for the XOR gate with $PT=0.6$. The physical QA distribution is shown in red. Despite getting closer to the ground state with these parameters, the results do not suggest a correlation with the physical QA results.

Longer trials will generally result in lower energy solutions, but those results will not necessarily be any more correlated with the results obtained from D-Wave.

The value of QMC as a simulator for QCA is not clear. It is possible that a set of parameters could be

found that would result in QMC simulating a physical QA distribution, but it is not apparent in this work.

6. CONCLUSIONS

This implementation of QMC generally succeeds in finding the ground state, but is consistently outperformed by SA. It does not replicate the success that both Martonak and Denchev report in their respective work in comparison against classical simulated annealing [1][2]. This suggests that this implementation is likely somewhat flawed and could almost certainly be improved.

Quantum Monte Carlo might become a useful simulation tool, but this implementation is not currently predictive. The information provided about the occupation of states does not map well to the physical quantum annealing data. Additionally, no pattern was observed relating simulation performance to any particular parameter selection. For QMC to function as a useful, predictive simulator for physical quantum annealing, it is necessary to develop a single method that achieves acceptable results for most problems.

REFERENCES

- [1] R. Martonak et al, “Quantum annealing by the path-integral Monte Carlo method: The two-dimensional random Ising model,” *Physical Review B* 66, 094203 (2002)
- [2] V. S. Denchev et al, “What is the Computational Value of Finite Range Tunneling?” arXiv:1512.02206v3 [quant-ph] (2015)
- [3] F. Barahona, “On the computational complexity of Ising spin glass models,” *J. Phys. A: Math. Gen.* 15: 3241-3253, 1982.
- [4] D-Wave Systems Inc. (2014). Applications [Online]. Available: <http://www.dwavesys.com/quantum-computing/applications> (Accessed: 24- Nov-2015).
- [5] C. S. Lent, “Quantum cellular automata,” *Nanotechnology*, 4:49–57, 1993.
- [6] J. Timler and C. S. Lent, “Power gain and dissipation in quantum-dot cellular automata,” *J. Appl. Phys.*, 91(2):823–831, January 2002.
- [7] C. S. Lent and B. Isaksen, “Clocked Molecular Quantum-Dot Cellular Automata,” *IEEE Trans. on Electron Dev.*, 50:1890–1896, 2003.
- [8] University of Cologne spin glass server, <http://www.informatik.uni-koeln.de/spinglass/>
- [9] K. Walus and G. A. Jullien, “Design tools for an emerging soc technology: quantum-dot cellular automata,” *Proc. IEEE*, 94(6):1225–1244, June 2006.

Additional thanks to Jake Retallick for his support in carrying out this research.