New Developments and Ideas at the Interface between Physics and Computer Science

Eduardo Mucciolo
Department of Physics
University of Central Florida
Overview

- Background and motivation: computation and physics
- Entanglement and irreversibility in circuits
- Ground-state computing and vertex models
- Tensor networks
Computer Science & Physics

The first thing to know:

“Computer science is no more about computers than astronomy is about telescopes.” (Edsger Dijkstra)

My take:

Computer science is a branch of mathematics that studies what is computable and how it can be computed.

But computations require real-world physical phenomena to be implemented!

Computer science and physics are closely connected.
Example: Landauer’s principle (1961)

Any logically irreversible manipulation of information, such as the erasure of a bit must be accompanied by a corresponding entropy increase.

\[ \Delta S \geq k_B \ln 2 \quad \rightarrow \quad \Delta E \geq k_B T \ln 2 \]

energy cost of bit erasure
But it is not only erasure that costs energy:

\[
\text{NAND gate}
\]

A bit has disappeared!

*Irreversibility in logic gates too!*

Actually, the connection between computational and irreversibility raised an interesting question:

Is it possible to compute reversibly?

Yes! (C. Bennett, 1973)

\[
\text{Toffoli gate (}\Delta E \geq 0)\]

*catch*: many more auxiliary bits are needed
There is another connection between physics and computer science:

Second connection: Physics provides models of computation

- Transistor-based circuit model (current standard)
  - Algorithm → set of arithmetic operations → set of logic gates → transistor operations

- Quantum-hardware based circuit model (q. computing)
  - Algorithm → set of quantum and classical abstract operations → quantum and classical hardware operations
There are other physical models of computation…

… including *ground-state models*


The result of the computation is encoded on the ground state of a physical system.

*Finding the ground state becomes the computation!*  

(1) Choose a system whose interactions enforce a pre-established algorithm.  

Two formulations:  

or  

(2) Choose a system whose interactions enforce the problem to be solved.
Example of (1): Enforcing a pre-established algorithm

\[ E_C(s_1, s_2, s_3, s_4, s_5) = \Delta(s_3 + s_4 + s_6 - s_3s_4 - 2s_3s_6 - 2s_4s_6 + 2s_3s_4s_6) \]

\[ + \Delta(s_5 + s_1s_2 - 2s_1s_2s_5) + \Delta(1 - s_7 - s_5s_6 + 2s_5s_6s_7). \]

\[ s_1, s_2, s_3, s_4 : \text{fixed} \]

\[ s_5, s_6, s_7 : \text{free} \]

When \( E_C = 0 \) \( \rightarrow \) \( s_5, s_6, s_7 : \text{computed} \) ✔
Example (2): Enforcing the problem to be solved

k-SAT problem (satisfiability of a conjunctive normal form):

\[ F(x_1, x_2, \ldots, x_N) = (\bar{x}_1 \lor x_2) \land (\bar{x}_2 \lor x_4 \lor x_5) \land (x_4 \lor \bar{x}_8 \lor \bar{x}_6) \land (x_7 \lor x_6 \lor x_2 \lor \bar{x}_3) \land (\bar{x}_3 \lor x_1) \]

- To each binary variable, associate a “bit” (e.g., a spin 1/2)
- To each clause, assign a many-body interaction term
- Add all interaction terms to create a total-energy cost function
- The ground state (lowest energy configuration) satisfies (or not!) all clauses simultaneously

\[ E_{\text{tot}} = E_{2-b}(x_1, x_2) + E_{2-b}(x_1, x_3) + E_{3-b}(x_2, x_4, x_5) + E_{3-b}(x_4, x_8, x_6) + E_{4-b}(x_7, x_6, x_2, x_3) \]
How can one reach the ground state?
**Simulated annealing**: slow cooling


slow cooling + thermal fluctuations
Quantum annealing: slow reshaping of the Hamiltonian

Apolloni, Carvalho, de Falco, Stochastic Process Appl. (1989)


Farhi, Goldstone, Gutmann, Laplan, Lundgren, Preda Science 292, 472 (2001)

D-Wave Systems

\[ H(t) = (1 - t/\tau)H_0 + (t/\tau)H_1 \]

adiabatic evolution
Numerical Simulation:
What is the main problem with these approaches?

Most ground-state encodings create physical systems that are too complex!

*They lead to glassy behavior and phase transitions!*

Both features slow down the annealing process…

*Our group may have found a way around (more on that soon)*
There is yet another connection between physics and computer science:

Third connection: Physics concepts can shed light on the complexity of computations.

Consider a black-box computation implemented with reversible gates.

\[
P
\begin{array}{c}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
\end{array}
\begin{array}{c}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8 \\
\end{array}
\]

How hard is it to reverse it?

Why is this an important question? Cryptography!

*Our group has found an answer (more on that soon as well)*
Our group has been developing applications of physics-inspired methods and ideas to computation:


- Chamon, Mucciolo, Ruckenstein & Yang, *Quantum vertex model for reversible classical computing* (Nat Comm, 2017)


_Boston-Orlando collaboration_
Irreversibility & Entanglement Spectrum in Reversible Circuits

- Consider a set of quantum bits (qubits): \( \{ |\psi_i\rangle \}_{i=1,\ldots,n} \)

- Start each one in a superposition state: \( |\psi_i\rangle = \sin(\theta_i) |0\rangle_i + \cos(\theta_i) |1\rangle_i \)

- The initial state of the system is a product state (no entanglement): \( |\Psi\rangle_{\text{initial}} = |\psi_1\rangle \otimes |\psi_2\rangle \cdots \otimes |\psi_n\rangle \)

- Apply the black-box circuit (a given sequence of reversible gates): \( U_{\text{circuit}} = U_m \cdot U_{m-1} \cdots U_1 \)

- The system will evolve into an entangled state: \( |\Psi_{\text{final}}\rangle = \sum_{x_1,\ldots,x_n} A(x_1,\ldots,x_n) |x_1\rangle \otimes \cdots \otimes |x_n\rangle \)

Is it possible to reverse the system back to a disentangled state, without referring to the circuit?
The answer: It depends on the type of circuit!

What we found:

For circuits containing a universal set of gates: NO
For circuits containing a non universal set of gates: YES

For classical reversible circuits, the Toffoli gate is universal:

For quantum circuits, Hadamard, CNOT, and T gates form a universal set:

\[ |0\rangle \rightarrow \mathbf{H} \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \]
\[ |1\rangle \rightarrow \mathbf{H} \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \]

\[ |0\rangle \rightarrow \mathbf{T} \rightarrow |0\rangle \]
\[ |1\rangle \rightarrow \mathbf{T} \rightarrow e^{i\pi/4} |1\rangle \]
How did we check reversibility and irreversibility?

**Trial and error:** We followed the entanglement entropy as we tried a random sequence of gates (Metropolis algorithm)

![Graphs showing entanglement entropy for different gate sets and conditions](image)

- **Non universal** gate sets: 
  - $\{\text{CNOT, H, NOT}\}$
  - $\{\text{CNOT, H, S}\}$
  - $\{\text{CNOT, H, T}\}$

- **Reversible** cases: 
  - (100 cases checked)

- **Irreversible** cases: 
  - (100 cases checked)
But what if we don’t know anything about the circuit? How could we tell?

The entanglement spectrum reveals the difficulty in reversing the evolution!

Subsystem A ($N_A$ qubits)
Subsystem B ($N_B$ qubits)

Entanglement spectrum statistics:

Ratio of consecutive spacings:

$$s_n = \frac{\lambda_n - \lambda_{n+1}}{\lambda_{n-1} - \lambda_n}$$

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \Rightarrow 0 \leq s_n < \infty$$

Distribution of ratios:

$$P(s) = \frac{1}{2^N_A} \left< \sum_{n=1}^{2^N_A} \delta(s - s_n) \right>$$

Reduced density matrix:

$$\rho_A = \text{tr}_B[\rho]$$

Eigenvectors:

$$\{\lambda_1, \lambda_2, \cdots, \lambda_{2^N_A}\}$$

Entanglement entropy:

$$S_1 = -\sum_{n=1}^{2^N_A} \lambda_i \ln \lambda_i$$

Constraint:

$$\lambda_n \geq 0$$

$$\sum_{n=1}^{2^N_A} \lambda_n = 1$$
**Results:** The quantum state of the system reveals the nature of the circuit!

But you need to look deep inside to find it.

- **reversible**
  \{CNOT, H, NOT\}
  - Poisson
  - GOE
  - GUE

- **reversible**
  \{CNOT, H, S\}
  - Poisson
  - GOE
  - GUE

- **irreversible**
  \{CNOT, H, T\}
  - Poisson
  - GOE
  - GUE

Wigner-Dyson statistics

5000 samples

$N_A = \frac{N}{2} = 8$
The statistics of the entanglement spectrum fluctuations reveal the nature of the circuit and the difficulty to reverse it.

\[
\begin{align*}
\text{Wigner-Dyson} & \quad \leftrightarrow \quad \text{irreversible} & \quad \leftrightarrow \quad \text{universal gates} \\
\text{Poisson} & \quad \leftrightarrow \quad \text{reversible} & \quad \leftrightarrow \quad \text{non universal gates}
\end{align*}
\]
Mapping Reversible Classical Circuits into Vertex Models

(A novel way to do ground-state computing)

- Create a two-dimensional lattice representation of a circuit; the gates are the vertices.
- One direction of the lattice represents “time”.
- The left and right boundaries contain the input and output states of the bits; can be fixed or free.
- Not limited to forward computations; can solve mixed-boundary problems.
How do we create the lattice?

Circuit building blocks: *reversible gates*
From gate to tile
From gate to tile

tilted for convenience
From circuits to tiled walls

All logic gates can be written using only 1- and 2-body interactions
From tile to vertex
Mapping reversible classical circuits to a vertex model

Other necessary gates: ID-ID, ID-SWAP, SWAP-ID, SWAP-SWAP

SWAP-SWAP

ID-ID
Mapping reversible classical circuits to a vertex model

Combining all gates (vertices) and couplings: a regular square lattice
Thermodynamics of the vertex model

$P_j : j$-th slice of reversible gates (permutation)

$T_j : j$-th slice of bit constraints (projector)

The exact partition function is computable!

**Result:** Paramagnet!

\[
Z = (2 \cosh \beta K)^{3LW}
\]

No finite-temperature phase transition!
Quantum vertex model: adding a transverse field

$2^3 = 8$ states at each vertex: $q_s = 0, 1, \ldots, 7$

$$
\hat{H} = \sum_{ss'} \sum_{q_s, q_{s'}} \left( K^s g_{s' q_s, q_{s'}} \right) |q_s q_{s'}\rangle \langle q_s q_{s'}|
+ \sum_{s \in \text{boundary}} \sum_{q_s} \hat{h}_{q_s} |q_s\rangle \langle q_s|
+ \sum_{s} \sum_{q_s, q_{s'}} \Delta_{q_s, q_{s'}} |q_s\rangle \langle q_s'|
,$$

quantum tunneling term

$J, h \gg K \gg \Delta$
The mapping onto the vertex model removes the finite-temperature glass transition.

Proposed phase diagram:

- **Thermal axis**: $\frac{T}{K}$
- **Quantum axis**: $\delta = \frac{\Delta}{K}$

- **No finite-$T$ transition**
- **Thermal annealing**

- **Local SAT**
- **Global SAT**
- **UNSAT**

$T \sim J$, $\Delta \sim J$, $\delta_c$
A solution can be reached… but how fast?

*Dynamics* is what matters!

The question to ask is:

How long does it take to thermalize?
Numerical investigation of thermal annealing dynamics

1) quasi-polynomial relax. times:

\[ \ell(\tau) = \ell_0 \exp\{[\ln(\tau/\tau_0)]^\gamma\} \]

\[ \gamma \approx 0.475 \quad \tau_{\text{sol}} \sim e(\#)(\ln L)^{2.1} \]

2) sub-exponential relax. times:

\[ \ell(\tau) = \ell_0 [\ln(\tau/\tau_0)]^\eta \]

\[ \eta \approx 1.69 \quad \tau_{\text{sol}} \sim e(\#) L^{0.6\theta} \ln L \]

\((0.6\theta < 1)\)

We cannot tell yet which one is the correct behavior!
Quantum annealing of the vertex model

No finite-\(T\) transition

thermal annealing

\(T \sim J\)

\(\frac{T}{K}\)

local SAT

global SAT

\(\delta_c\)

\(\delta = \frac{\Delta}{K}\)

\(\Delta \sim J\)

quantum annealing

Is there a quantum phase transition?
Quantum annealing of the vertex model

- When no Toffoli gates are presented: equivalent to 1D Ising chains + transverse field

  Only a 2nd order quantum phase transition

  annealing in polynomial time!

- When a finite density of Toffoli gates are presented: the same scenario is likely. *(conjecture)*

- Ongoing Quantum Monte Carlo studies to check it.

- Plans to verify it experimentally using a quantum annealer: D-Wave machine
Embedding into the D-Wave Chimera Architecture

Figure 14. Procedure for embedding a 4\times4 tile lattice into the Chimera graph. (a) Left: a generic tile lattice rotated by 45\degree. Spins are put on the boundary of each tile. The lattice can be further divided into two sublattices, depicted by dark and light grey respectively; right: embedding of the tile lattice into the Chimera graph. The “grout couplings” are indicated by red links. (b) Embedding of each gate into the unit cells of the Chimera graph. (i) Left: a $K_4^4$, $4\times4$ unit cell of the Chimera graph; middle: in order to couple qubits in the same column, we slave the qubits to their neighbors in the other column using additional ferromagnetic couplings indicated by red links; right: effectively we are left with four qubits that are fully connected. For simplicity, we hereafter denote the effective couplings between spins in the same column by a single green link. However, one should keep in mind that they are obtained by slaving the spins to the opposite column via large ferromagnetic couplings. (ii) The four qubits in the rotated square tile are labeled by their locations on the tile: N (North), S (South), W (West) and E (East). Tiles corresponding to different sublattices must be embedded differently due to the special connectivity of the Chimera graph. (iii) Embedding of the TOFFOLI gate consisting of two square tiles into two unit cells. (a, b, c, d) corresponds to the input and output bits of the gate, and $S$ is the ancilla bit. In the unit cell, ferromagnetic couplings that copy spins are indicated by purple links, and couplings required in Hamiltonian (3) are indicated by black links.

The quantum vertex model can be implemented in the D-Wave machine.
Tensor Network Approach to Vertex Models

Gates $\rightarrow$ Tensors (vertices)

$k = F_1(i, j)$
$l = F_2(i, j)$

$$T_{i,j,k,l} = \begin{cases} 
1, & \text{if } k = F_1(i, j) \text{ and } l = F_2(i, j), \\
0, & \text{otherwise}.
\end{cases}$$
tensors $\rightarrow$ network

boundary indices: fixed or free, depending on the problem to be solved
Instead of looking for ground states, we take a different approach:

\[ Z = \text{Tr} \prod_n T[n]_{i,j,k,l} \quad \text{full contraction of the tensor network} \]

Example:

\[ Z(\alpha, \beta) = \text{Tr} \begin{bmatrix} 1 & 2 & \beta \\ i & j & \alpha \\ n & k & m \end{bmatrix} = \sum_{i,j,k,l,m,n} T[1]_{i,j,n} T[2]_{j,\beta,l} T[3]_{\alpha,n,m} T[4]_{k,l,m} \]

\[ Z(\alpha, \beta) = \text{number of solutions to the problem encoded by the circuit for fixed } \alpha, \beta. \]
By varying boundary variables, one at a time, and monitoring $Z$, we can determine solutions to the problem.

The catch: it is exponentially hard to compute $Z$!! \textit{\#P complexity class}

$\text{computational cost } \sim O \left( \chi \# \text{ of network links} \right)$

\text{bond dimension}

We have developed a scheme that attenuates the difficult:

Iterative Compression Decimation (ICD)
Compression of every bond

Decimation of rows and columns
Some recent numerical results for the random Toffoli circuit:

We were able to reach $L = 96$.

A brute-force enumeration would require $8^{48} \sim 10^{43}$ iterations!!
SUMMARY:

- Close connections between Computer Science & Physics

- Physics constraints computations, but also inspires new methods and concepts

- Entanglement spectrum fluctuations reveal the difficult to reverse circuits

- Vertex models: a novel way to do ground-state computing (classical or quantum)

- ICD: A new scheme to solve vertex-model problems using tensor networks
Collaborators:

Claudio Chamon
Andrei Ruckenstein
Zhicheng Yang
Alioscia Hamma
Sabine Pelton
Justin Reyes

Stefanos Kourtis
Daniel Shaffer
Lei Zhang
Dan Marinescu
Pawel Wocjan
Partial Funding:
THE END