Welcome

2017 Faculty Summit Attendees

Faculty Summit 2017
microsoftfacultysummit.com

Microsoft Research
Microsoft.com/research

Facebook
@microsoftresearch

Twitter
@MSFTResearch
#FacSumm
#EdgeofAI
Quantum vs Classical Optimization: A status update on the arms race

Helmut G. Katzgraber
https://intractable.lol
Quantum vs Classical Optimization: A status update on the arms race

Helmut G. Katzgraber
https://intractable.lol
Quantum vs Classical Optimization: A status update on the arms race

Helmut G. Katzgraber
https://intractable.lol
• Some questions we would like answers to…
  • Current status of quantum vs classical optimization?
  • What about quantum approaches for machine learning?
  • If QA fails to deliver, can we still benefit?

• Texas A&M team:
Some questions we would like answers to…

- Current status of quantum vs classical optimization?
- What about quantum approaches for machine learning?
- If QA fails to deliver, can we still benefit? Think quantum inspired…

Texas A&M team:

as well as… S. Mandrà @ NASA, F. Hamze @ D-Wave, C. Thomas @ Google.
Some questions we would like answers to…

- Current status of quantum vs classical optimization?
- What about quantum approaches for machine learning?
- If QA fails to deliver, can we still benefit? Think quantum inspired…

Texas A&M team:

- S. Mandrà
- F. Hamze
- C. Thomas
- C. Fang
- Dr. W. Wang
- J. Chancellor
- Dr. Z. Zhu
- A. Barzegar
- A. Ochoa
- C. Pattison missing

as well as… S. Mandrà @ NASA, F. Hamze @ D-Wave, C. Thomas @ Google.
Why quantum annealing? Optimization!

- Selected problems of interest:
  - Constraint satisfaction (SAT)
  - Number partitioning
  - Minimum vertex covers
  - Traveling salesman problem, ...

- What do all these have in common?
  - Rough cost function landscapes.
  - They are problems in NP (also typical hard).
  - All map onto Quadratic Unconstrained Binary Optimization (QUBO) problems.

\[
\mathcal{H}(S_i) = \sum_{i \neq j}^{N} Q_{ij} S_i S_j \\
S_i \in \{\pm 1\}
\]
Why quantum annealing? Optimization!

• Selected problems of interest:
  • Constraint satisfaction (SAT)
  • Number partitioning
  • Minimum vertex covers
  • Traveling salesman problem, …

• What do all these have in common?
  • Rough cost function landscapes.
  • They are problems in NP (also typical hard).
  • All map onto Quadratic Unconstrained Binary Optimization (QUBO) problems.

\[ H(S_i) = \sum_{i \neq j} Q_{ij} S_i S_j \quad S_i \in \{ \pm 1 \} \]
Why quantum annealing? Optimization!

- Selected problems of interest:
  - Constraint satisfaction (SAT)
  - Number partitioning
  - Minimum vertex covers
  - Traveling salesman problem, …

- What do all these have in common?
  - Rough cost function landscapes.
  - They are problems in NP (also typical hard).
  - All map onto Quadratic Unconstrained Binary Optimization (QUBO) problems.

\[
\min_{S_i} \mathcal{H}(S_i) = \sum_{i \neq j}^N Q_{ij} S_i S_j \quad S_i \in \{\pm 1\}
\]
Why quantum annealing? Optimization!

• Selected problems of interest:
  • Constraint satisfaction (SAT)
  • Number partitioning
  • Minimum vertex covers
  • Traveling salesman problem, …

• What do all these have in common?
  • Rough cost function landscapes.
  • They are problems in NP (also typical hard).
  • All map onto Quadratic Unconstrained Binary Optimization (QUBO) problems.

\[
\min_{S_i} \mathcal{H}(S_i) = \sum_{i \neq j}^{N} Q_{ij} S_i S_j
\]
Moore’s Law is coming to an end...

- Four possible ways to overcome the end of Moore’s law:
  - Build larger silicon-based computers.
  - Develop faster silicon-based technologies.
  - Focus on faster algorithms.
  - Go beyond standard silicon architectures.

- Here, deep synergy between...
  - Physics,
  - ...quantum information, ...
  ... and computer science.

![Graph showing transistor count over time](adapted from Nature (2016))
Moore’s Law is coming to an end…

- Four possible ways to overcome the end of Moore’s law:
  - Build larger silicon-based computers.
  - Develop faster silicon-based technologies.
  - Focus on faster algorithms.
  - Go beyond standard silicon architectures.

- Here, deep synergy between…
  - Physics,…
  - …quantum information,… … and computer science.
Moore’s Law is coming to an end…

- Four possible ways to overcome the end of Moore’s law:
  - Build larger silicon-based computers.
  - Develop faster silicon-based technologies.
  - Focus on faster algorithms.
  - Go beyond standard silicon architectures.

- Here, deep synergy between…
  - Physics,…
  - …quantum information,…
  - … and computer science.
Moore’s Law is coming to an end…

• Four possible ways to overcome the end of Moore’s law:
  • Build larger silicon-based computers. already close to fab limits
  • Develop faster silicon-based technologies.
  • Focus on faster algorithms.
  • Go beyond standard silicon architectures.

• Here, deep synergy between…
  • Physics,…
  • …quantum information, … … and computer science.
Moore’s Law is coming to an end…

- Four possible ways to overcome the end of Moore’s law:
  - Build larger silicon-based computers. already close to fab limits
  - Develop faster silicon-based technologies. potentially disruptive
  - Focus on faster algorithms.
  - Go beyond standard silicon architectures. not scalable

- Here, deep synergy between…
  - Physics,…
  - …quantum information, …
  - … and computer science.

![Graph showing transistor count vs. clock speed from 1970 to 2010, adapted from Nature (2016).]
Moore’s Law is coming to an end...

- Four possible ways to overcome the end of Moore’s law:
  - Build larger silicon-based computers. (already close to fab limits)
  - Develop faster silicon-based technologies. (potentially disruptive)
  - Focus on faster algorithms.
  - Go beyond standard silicon architectures.

- Here, deep synergy between...
  - Physics,…
  - …quantum information, …
  - … and computer science.

![Graph showing transistor count vs. time from 1970 to 2010](Image 24960x658 to 25746x729)
Current state of the art:

*Special-purpose analog quantum annealers*

Antikythera ~ 80BC
• What is it?
  • Semi-programmable analog annealer.
  • 2000 superconducting flux qubits.
  • Controversial performance.
  • Still, huge technological feat…

• What can it do?
  • It can minimize QUBOs post embedding onto the machine’s hardwired Chimera topology.

• Limitations:
  • Low connectivity.
  • Analog noise.
  • …
• **What is it?**
  - Semi-programmable analog annealer.
  - 2000 superconducting flux qubits.
  - Controversial performance.
  - Still, huge technological feat…

• **What can it do?**
  - It can minimize QUBOs post embedding onto the machine’s hardwired Chimera topology.

• **Limitations:**
  - Low connectivity.
  - Analog noise.
  - …
• What is it?
  • Semi-programmable analog annealer.
  • 2000 superconducting flux qubits.
  • Controversial performance.
  • Still, huge technological feat…

• What can it do?
  • It can minimize QUBOs post embedding onto the machine’s hardwired Chimera topology.

• Limitations:
  • Low connectivity.
  • Analog noise.
  • …
How do quantum annealers optimize?
How do quantum annealers optimize?

Sequentially.
Classical Analog: Simulated Annealing (SA)

- Annealing:
  - 7000 year-old neolithic technology.
  - Slowly cool to remove imperfections.

- Simulated Annealing (SA):
  - Stochastically sample $\mathcal{H}(\{S\})$ using Monte Carlo.
  - If the system is thermalized, cool it.
  - The slower the cooling, the better, e.g.,
    
    $$T(t) = a - bt$$

- Problem: SA is inefficient for complex systems.
- Solution: Multiple restarts & statistics gathering.
Classical Analog: Simulated Annealing (SA)

- **Annealing:**
  - 7000 year-old neolithic technology.
  - Slowly cool to remove imperfections.

- **Simulated Annealing (SA):**
  - Stochastically sample $\mathcal{H}(\{S\})$ using Monte Carlo.
  - If the system is thermalized, cool it.
  - The slower the cooling, the better, e.g.,

\[
T(t) = a - bt
\]

- **Problem:** SA is inefficient for complex systems.
- **Solution:** Multiple restarts & statistics gathering.

*German copper axe*

Kirkpatrick et al., Science (83)

Geman & Geman

$T(t) = a - bt$
Classical Analog: Simulated Annealing (SA)

Kirkpatrick et al., Science (83)

• **Annealing:**
  - 7000 year-old neolithic technology.
  - Slowly cool to remove imperfections.

• **Simulated Annealing (SA):**
  - Stochastically sample $\mathcal{H}(\{S\})$ using Monte Carlo.
  - If the system is thermalized, cool it.
  - The slower the cooling, the better, e.g.,
  \[
  T(t) = a - bt
  \]

• **Problem:** SA is *inefficient* for complex systems.
• **Solution:** Multiple restarts & statistics gathering.
Classical Analog: Simulated Annealing (SA)

- **Annealing:**
  - 7000 year-old neolithic technology.
  - Slowly cool to remove imperfections.

- **Simulated Annealing (SA):**
  - Stochastically sample $\mathcal{H}(\{S\})$ using Monte Carlo.
  - If the system is thermalized, cool it.
  - The slower the cooling, the better, e.g.,
    \[
    T(t) = a - bt
    \]

- **Problem:** SA is *inefficient* for complex systems.
- **Solution:** Multiple restarts & statistics gathering.

*Kirkpatrick et al., Science (83)*

*Geman & Geman*
Quantum Annealing (QA)

• **Idea:**
  - Use quantum fluctuations instead of thermal.
  - Sequential algorithm like SA.

• **Theoretical advantages over SA:**
  - Fluctuations determine the “tunneling radius.”
  - Not limited to a local search.

• **Implementation in DW device (transverse-field QA):**
  - Apply a *transverse field* that does not commute: 
    \[ [S^x, S^z] \neq 0 \]
    \[
    \mathcal{H}(S_i) = \sum_{i \neq j} Q_{ij} S_i S_j \quad \rightarrow \quad \mathcal{H}(S_i) = \sum_{i \neq j} Q_{ij} S_i^z S_j^z - D \sum_i S_i^{x}
    \]
  - Reduce the fluctuation amplitude $D$ via a given annealing protocol.
Quantum Annealing (QA)

• Idea:
  • Use quantum fluctuations instead of thermal.
  • Sequential algorithm like SA.

• Theoretical advantages over SA:
  • Fluctuations determine the “tunneling radius.”
  • Not limited to a local search.

• Implementation in DW device (transverse-field QA):
  • Apply a transverse field that does not commute:
    \[ [S^x, S^z] \neq 0 \]
    \[
    \mathcal{H}(S_i) = \sum_{i \neq j} Q_{ij} S_i S_j \quad \rightarrow \quad \mathcal{H}(S_i) = \sum_{i \neq j} Q_{ij} S_i^z S_j^z - D \sum_i S_i^x
    \]
  • Reduce the fluctuation amplitude \( D \) via a given annealing protocol.
Promising signs of quantum speedup...?

see Mandrà, Zhu, Perdomo-O. & Katzgraber (PRA, arXiv:1604.01746)
Google’s “$10^8$ results” – slope vs offset

Denchev et al. (15)

FIG. 4. Time to find the optimal solution with 99% probability (TTS) in $\mu$s, as a function of problem size $N$. For each algorithm, we plot the 50th, 75th, and 85th percentiles of the time to success (TTS). The figure shows that D-Wave has a significantly shorter TTS compared to Simulated Annealing (SA) and Quantum Monte Carlo (QMC), especially for larger problem sizes. The D-Wave algorithm is able to find the optimal solution much faster than the classical algorithms, highlighting its potential for solving complex optimization problems.

For the classical algorithms, we calculate the total computational effort required to reach a 99% success probability as $N^2 \ln \frac{1}{\eta}$, where $N$ is the number of spin updates (for SA) or worldline updates (for QMC) that are required to reach a 99% success probability. Shown are the 50th, 75th and 85th percentiles for the largest problem size for QMC were multiplied that with the time to perform one update on different parameters please.
Google’s “10^8 results” – slope vs offset

Denchev et al. (15)
Google’s “10^8 results” – slope vs offset

Denchev et al. (15)
Google’s “10^8 results” – slope vs offset

Denchev et al. (15)

\[ \mathcal{H}(S_i) = \sum_{i \neq j}^{N} Q_{ij} S_i S_j - \sum_{i}^{N} h_i S_i \]
Google’s “10^8 results” – slope vs offset

Denchev et al. (15)

spin-glass backbone

\[ H(S_i) = \sum_{i \neq j}^{N} Q_{ij} S_i S_j - \sum_{i} h_i S_i \]
Google’s “10^8 results” – slope vs offset

Denchev et al. (15)
Google’s “$10^8$ results” – slope vs offset

Denchev et al. (15)
Google’s “10^8 results” – slope vs offset

Denchev et al. (15)

Better scaling of DW and quantum inspired.
\( h = 0 \quad h > 0 \)

\begin{align*}
0 & \quad 1
\end{align*}
What if we use better algorithms?

- Tailored to the problems and/or underlying graph:
  - Hamze-de Freitas-Selby algorithm (HFS). [Hamze et al. (12)]
  - Hybrid cluster methods (HCM). [Venturelli, et al. (15)]
  - Super-spin approximation (SS). [Zhu (16)]

- Not tailored to the problems and/or underlying graph:
  - Population annealing (particle swarm) sequential Monte Carlo (PA). [Wang et al., PRE (15)]
  - Parallel tempering & isoenergetic cluster optimizer (PT+ICM). [Zhu, Ochoa, Katzgraber PRL (15)]

- Reminder – Sequential methods used in the Google study:
  - Simulated annealing (SA). [Kirkpatrick et al. (83)]
  - Quantum Monte Carlo (QMC). [Denchev et al. (15)]
  - D-Wave 2X (DW2).
What if we use better algorithms?

- Tailored to the problems and/or underlying graph:
  - Hamze-de Freitas-Selby algorithm (HFS). \textit{Hamze et al. (12)}
  - Hybrid cluster methods (HCM). \textit{Venturelli, et al. (15)}
  - Super-spin approximation (SS). \textit{Zhu (16)}

- Not tailored to the problems and/or underlying graph: \textit{Wang et al., PRE (15)}
  - Population annealing (particle swarm) sequential Monte Carlo (PA).
  - Parallel tempering & isoenergetic cluster optimizer (PT+ICM). \textit{Zhu, Ochoa, Katzgraber PRL (15)}

- Reminder – Sequential methods used in the Google study:
  - Simulated annealing (SA). \textit{Kirkpatrick et al. (83)}
  - Quantum Monte Carlo (QMC). \textit{Denchev et al. (15)}
  - D-Wave 2X (DW2).
What if we use better algorithms?

- Tailored to the problems and/or underlying graph:
  - Hamze-de Freitas-Selby algorithm (HFS). 
    Hamze et al. (12)
  - Hybrid cluster methods (HCM). 
    Venturelli, et al. (15)
  - Super-spin approximation (SS). 
    Zhu (16)

- Not tailored to the problems and/or underlying graph: 
  Wang et al., PRE (15)
  - Population annealing (particle swarm) sequential Monte Carlo (PA).
  - Parallel tempering & isoenergetic cluster optimizer (PT+ICM). 
    Zhu, Ochoa, Katzgraber PRL (15)

- Reminder – Sequential methods used in the Google study:
  - Simulated annealing (SA). Kirkpatrick et al. (83)
  - Quantum Monte Carlo (QMC). Denchev et al. (15)
  - D-Wave 2X (DW2).
What if we use better algorithms?

- Tailored to the problems and/or underlying graph:
  - Hamze-de Freitas-Selby algorithm (HFS). \(\text{Hamze et al. (12)}\)
  - Hybrid cluster methods (HCM). \(\text{Venturelli, et al. (15)}\) \(\text{Zhu (16)}\)
  - Super-spin approximation (SS).

- Not tailored to the problems and/or underlying graph: \(\text{Wang et al., PRE (15)}\)
  - Population annealing (particle swarm) sequential Monte Carlo (PA).
  - Parallel tempering & isoenergetic cluster optimizer (PT+ICM). \(\text{Zhu, Ochoa, Katzgraber PRL (15)}\)

- Reminder – Sequential methods used in the Google study:
  - Simulated annealing (SA). \(\text{Kirkpatrick et al. (83)}\)
  - Quantum Monte Carlo (QMC). \(\text{Denchev et al. (15)}\)
  - D-Wave 2X (DW2).
Asymptotic scaling exponent $b$ (slope)

\[ T \sim \text{poly}(\sqrt{n})10^{a+b\sqrt{n}} \]
Asymptotic scaling exponent $b$ (slope)

$T \sim \text{poly}(\sqrt{n})10^{a+b\sqrt{n}}$

$b [50\%, \text{main scaling exponent}]$

smaller means better scaling

SA  PA  DW2  QMC  HCM  RMC+ICM  PT+ICM  HFS  SS
Asymptotic scaling exponent $b$ (slope)

$$T \sim \text{poly}(\sqrt{n})10^{a+b\sqrt{n}}$$

$b$ [50%, main scaling exponent]

<table>
<thead>
<tr>
<th>Method</th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DW2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMC+ICM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT+ICM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sequential: tailored

Not tailored

Tailored

Smaller means better scaling
Asymptotic scaling exponent $b$ (slope)

$T \sim \text{poly}(\sqrt{n})10^{a+b\sqrt{n}}$

Only “sequential” quantum speedup.
Most recent D-Wave benchmarks

see Mandrà, Katzgraber & Thomas (QST, arXiv:1703.00622)
D-Wave’s frustrated cluster loop problems

\[ \alpha = 0.80, \quad \rho = 5 \]

Number of logical variables

MWPM (no broken qubits)

TTS (\(\mu s\))

King et al. (17)
D-Wave’s frustrated cluster loop problems

\[ \alpha = 0.80, \ \rho = 5 \]

Number of logical variables

MWPM (no broken qubits)

MWPM
dW2000Q, TTS

ICM (logical), TTS

TTS \([\mu s]\)

King et al. (17)

\( \sqrt{n} \) [number of logical variables]
D-Wave’s frustrated cluster loop problems

- Ruggedness of FCLs (spin-glass backbone) fools codes.
- The logical problem is defined on $K_{44}$ cells and is therefore planar.

King et al. (17)

\[\alpha = 0.80, \quad \rho = 5\]

TTS (\(\mu s\))

Number of logical variables

MWPM (no broken qubits)

MWPM

DW2000Q, TTS

ICM (logical), TTS

Catapult + QMC

SA

DW2000Q

$\sqrt{n}$ [number of logical variables]
D-Wave’s frustrated cluster loop problems

- Ruggedness of FCLs \((\text{spin-glass backbone})\) fools codes.
- The logical problem is defined on \(K_{44}\) cells and is therefore planar.

Why is this a problem?
- Planar problems are polynomial (P class).
- Exact algorithms exist.

King et al. (17)
Using minimum-weight perfect perfect matching...

TTS (µs) vs. √n [number of logical variables]

- King et al. (17)
- Mandrà et al. (17)
- Edmonds (61)

Number of logical variables

$\alpha = 0.80$, $\rho = 5$ mwpm (fully-chimera) $\mu$mwpm, log(0.01)/log(1-p) mwpm
Using minimum-weight perfect matching...

\[ TTS (\mu s) \]

\[ \alpha = 0.80, \quad \rho = 5 \]

\[ mwpm (fully-chimera) \]

\[ DW2000Q, 1/p \]

\[ DW2kQ, \log(0.01)/\log(1-p) \]

Using minimum-weight perfect matching…

\[ TTS [\mu s] \]

\[ \sqrt{n} [\text{number of logical variables}] \]
Using minimum-weight perfect matching...

Exponentially faster than DW2000Q...

King et al. (17)

Mandrà et al. (17)

Edmonds (61)

\( \alpha = 0.80, \rho = 5 \)

MWPM

\[ \text{TTS} \left( \mu s \right) \]

\( \sqrt{n} \) [number of logical variables]
Using minimum-weight perfect matching…

Using minimum-weight perfect matching…

Exponentially faster than DW2000Q…

\[ \sqrt{n} \] [number of logical variables]

TTS [µs]

DW2000Q

King et al. (17)

Mandrà et al. (17)

Edmonds (61)

MWPM
\( h = 0 \)

\( h > 0 \)

2 : 1
Fair sampling – A key ingredient in ML

see also Mandrà, Zhu & Katzgraber (PRL, arXiv:1606.07146)
What is fair sampling?

- **Definition (fair sampling):**
  - Ability of an algorithm to find uncorrelated solutions to a problem with (almost) the same probability.

- **Why is this important?**
  - Sometimes solutions are more important than the optimum (SAT filters, #SAT, machine learning,…).
  - Some solutions might be more “convenient” due to additional constraints.

- **Algorithm benchmarking:**
  - **Standard** — Find the optimum *fast* and *reliably*.
  - **Stringent** — Find *all* minimizing configurations *equiprobably*. 
What is fair sampling?

**Definition (fair sampling):**
- Ability of an algorithm to find uncorrelated solutions to a problem with (almost) the same probability.

**Why is this important?**
- Sometimes solutions are more important than the optimum (SAT filters, #SAT, machine learning,…).
- Some solutions might be more “convenient” due to additional constraints.

**Algorithm benchmarking:**
- Standard  — Find the optimum *fast* and *reliably*.
- Stringent  — Find *all* minimizing configurations *equiprobably*. 
What is fair sampling?

• Definition (fair sampling):
  • Ability of an algorithm to find uncorrelated solutions to a problem with (almost) the same probability.

• Why is this important?
  • Sometimes solutions are more important than the optimum (SAT filters, #SAT, machine learning,…).
  • Some solutions might be more “convenient” due to additional constraints.

• Algorithm benchmarking:
  • Standard — Find the optimum fast and reliably.
  • Stringent — Find all minimizing configurations equiprobably.

*current state of the art is PT+ICM*
Can transverse-field QA sample fairly?

- 5-variable toy model suggests bias:
  - $J_{ij} = +1$
  - $J_{ij} = -1$
  \[ \mathcal{H} = \sum_{\langle ij \rangle} J_{ij} S_i S_j \]

- What about quantum annealers?
  - Design problems with known degeneracy:
    \[ \mathcal{H} = \sum_{\langle ij \rangle} J_{ij} S_i S_j \quad J_{ij} \in \{ \pm 5, \pm 6, \pm 7 \} \rightarrow \quad N_{GS} = 3 \cdot 2^k, \quad k \in \mathbb{N} \]
  - Study the distribution of ground states for fixed $N_{GS}$.

Matsuda, Nishimori, Katzgraber (NJP 2009)
Can transverse-field QA sample fairly?

- 5-variable toy model suggests bias:
  \[ H = \sum_{\langle ij \rangle} J_{ij} S_i S_j \]
  \[
  J_{ij} = +1 \\
  J_{ij} = -1
  \]

- What about quantum annealers?
  - Design problems with known degeneracy:
    \[ H = \sum_{\langle ij \rangle} J_{ij} S_i S_j \]
    \[ J_{ij} \in \{ \pm 5, \pm 6, \pm 7 \} \]
    \[ N_{GS} = 3 \cdot 2^k, k \in \mathbb{N} \]
  - Study the distribution of ground states for fixed \( N_{GS} \).
Can transverse-field QA sample fairly?

- 5-variable toy model suggests bias:

\[
\begin{align*}
\mathcal{H} &= \sum_{\langle ij \rangle} J_{ij} S_i S_j \\
J_{ij} &= +1 \\
J_{ij} &= -1
\end{align*}
\]

- What about quantum annealers?

  - Design problems with known degeneracy:
    \[
    \mathcal{H} = \sum_{\langle ij \rangle} J_{ij} S_i S_j \quad J_{ij} \in \{ \pm 5, \pm 6, \pm 7 \} \quad N_{\text{GS}} = 3 \cdot 2^k, k \in \mathbb{N}
    \]
  
  - Study the distribution of ground states for fixed \( N_{\text{GS}} \).
Can transverse-field QA sample fairly?

- 5-variable toy model suggests bias:

\[ H = \sum_{\langle ij \rangle} J_{ij} S_i S_j \]

\( J_{ij} = +1 \)
\( J_{ij} = -1 \)

- What about quantum annealers?
  - Design problems with known degeneracy:

\[ H = \sum_{\langle ij \rangle} J_{ij} S_i S_j \quad J_{ij} \in \{ \pm 5, \pm 6, \pm 7 \} \rightarrow N_{GS} = 3 \cdot 2^k, k \in \mathbb{N} \]

- Study the distribution of ground states for fixed \( N_{GS} \).
Transverse-field QA is exponentially biased

Transverse-field QA is exponentially biased.
Transverse-field QA is exponentially biased

Sample data for $N = 684$

Standard QA will need tweaks for fair sampling.
\( h = 0 \)

\( h > 0 \)

2:1
\[ h = 0 \quad \text{vs.} \quad h > 0 \]

3:1
$h = 0 \quad h > 0$

3 : 1

analog QA
Look out for IARPA’s QEO report on QA.

$h = 0$ vs $h > 0$

3:1 analog QA
Look out for IARPA’s QEO report on QA.
However… Soon superseded by digital?
Quantum vs Classical Optimization: A status update on the arms race

- Classical optimization pushes quantum technology.
- Quantum developments leverage classical quantum inspired methods.
- ML could benefit from quantum samplers... if these can sample fairly.
- To date, no application speedup or better scaling of quantum annealing.

contact-us@intractable.lol
Quantum vs Classical Optimization: A status update on the arms race

Thank you.

- Classical optimization pushes quantum technology.
- Quantum developments leverage classical quantum inspired methods.
- ML could benefit from quantum samplers... if these can sample fairly.
- To date, no application speedup or better scaling of quantum annealing.

contact-us@intractable.lol
Thank you