Snell Tomography for Net-To-Gross Estimation Using Quantum Annealing

Rahul Sarkar[1] Stewart A. Levin[2]

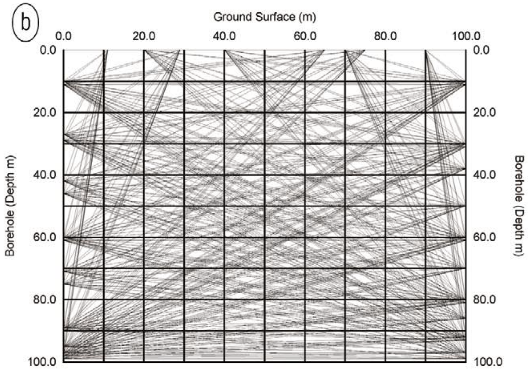
[1]Institute for Computational and Mathematical Engineering (ICME)
Stanford University

[2]Department of Geophysics
Stanford University

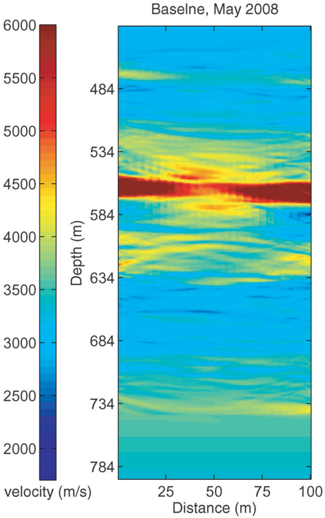
SEG 2018 Annual Meeting, Anaheim

Ray Tomography

Tomography ≈ localizing subsurface properties



A.Curtis (2004)



Zhang et al. (2012)

Ray Tomography

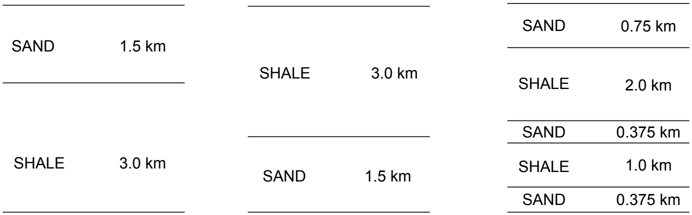
Tomography $\approx$ localizing subsurface properties

But not always...

Total Ambiguity

1 Ray, 2 Materials

- Sand – 3.0 km/s
- Shale – 2.5 km/s
- Experiment vertical ray
  - Source-receiver distance = 4.5 km
  - Source-receiver traveltime = 1.7 s

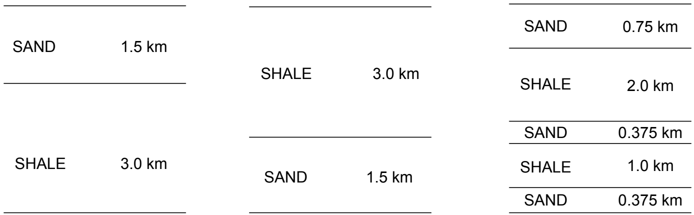


Many answers

No Ambiguity

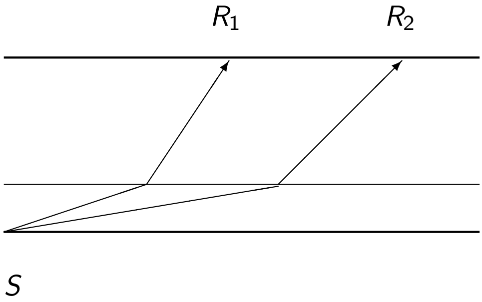
1 Ray, 2 Materials

- Sand – 3.0 km/s
- Shale – 2.5 km/s
- Experiment vertical ray
  - Source-receiver distance = 4.5 km
  - Source-receiver traveltime = 1.7 s



Exactly one sand-shale ratio (1:2)!

Challenge: *Can we determine the fraction of each material in the macrolayer between the source and receivers?*

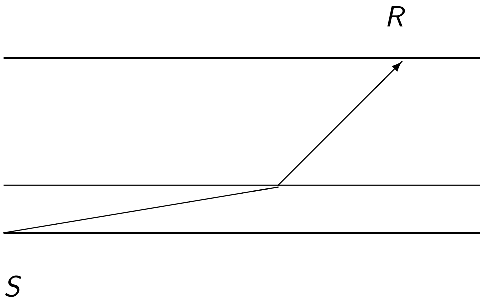


Experimental Geometry

1D Earth Model

- Horizontal layers of unknown thicknesses but known material properties
- The receivers are placed on the top of the layers
- There is one source at the bottom of the layers
- We measure both **travel time** and **offset** for every source-receiver pair

OMG!



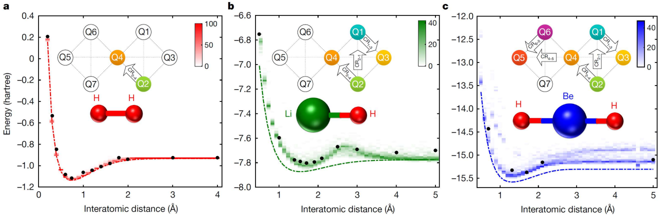
$$
\begin{aligned}
0 \quad = \quad & 4\gamma^2(\gamma^2-1)^2 T^2 v_1^2 (Z^2+L^2)\, s^6 \\
& -4\gamma(\gamma^2-1)^2 T v_1 L[(\gamma^2+1)(L^2+Z^2)+T^2 v_1^2]\, s^5 \\
& +(\gamma^2-1)^2[(\gamma^2-1)^2 Z^4 + 2(\gamma^4+1)L^2 Z^2 + (\gamma^2+1)^2 L^4 \\
& \qquad\qquad +2(\gamma^2+1)T^2 v_1^2 (L^2-Z^2) + T^4 v_1^4]\, s^4 \\
& +4\gamma(\gamma^2-1)^2 T v_1 L(L^2+2Z^2)\, s^3 \\
& -2(\gamma^2-1)^2 L^2[(\gamma^2+1)(L^2+Z^2)+T^2 v_1^2]\, s^2 \\
& +(\gamma^2-1)^2 L^4
\end{aligned}
$$

Potential of Quantum Computing

Some problems that are hard to solve using today's classical computers, have the potential to be efficiently solvable using a quantum computer.

Some possible use cases of quantum computing:

- Large combinatoric optimization problems
- Quantum chemistry & material design
- Quantum cryptography
- Quantum machine learning



Kandala et al., Nature 549 (2017)

Quo Vadis Quantum Computing?

Commercial general purpose quantum computers are 5-10 years out.

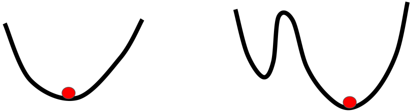
- Today: $<$ 100 quantum bits
- Near term: $\sim$ 1000 quantum bits (NISQ-era)
- Commercial: $\sim$ 1 million quantum bits

Specialized devices called quantum annealers exist today.

- Today: 2000 quantum bits
- Near Term: $\sim$ 5000–10000 quantum bits

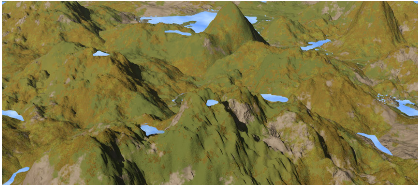
What Are Quantum Annealers?

- Specialized quantum devices
- Finds the minimum energy eigenstate for Ising Hamiltonians



$$H_t = (1 - \frac{t}{T})H_i + \frac{t}{T}H_f$$

Adiabatic quantum computing: If you change slowly the initial Hamiltonian to the final Hamiltonian, the system will remain in its lowest energy state.



www.dwavesys.com/quantum-computing

Quadratic Unconstrained Binary Optimization (QUBO)

One possible use of quantum annealers, is to solve QUBO problems!

Model QUBO Problem

$$\begin{aligned} \text{minimize} \quad & x^T Q \, x \\ \text{subject to} \quad & x \in \{0, 1\}^n \end{aligned}$$

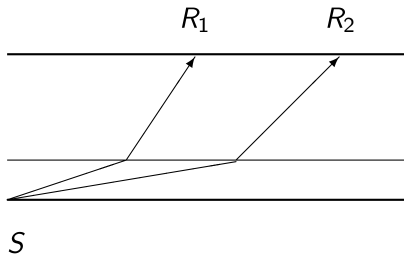
- Many other problems can be reduced to the QUBO form, for example problems with linear equality and inequality constraints.
- In our problem $\mathbf{x}$ assigns materials to layers to match traveltimes and offsets.

Problem Formulation

Horizontal constant thickness layers : We will assume that we have a finite number N of horizontal sublayers with same thickness δ. So total thickness is $N\delta$.

Finite material set : There are only a finite number of materials M, with **fixed known velocities** given by $v_1, \ldots, v_M$.

Finite number of rays : We have a finite number of rays with corresponding travel time measurements are given by $T_1, \ldots, T_K$, and the offsets are given by $L_1, \ldots, L_K$.



Layer Assignment

Introduce binary variables :

$$x_{ij} = \begin{cases} 1 & \text{if sublayer } i \text{ is material } j \\ 0 & \text{otherwise,} \end{cases}$$

for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$.

Introduce constraints :

To ensure that each sublayer gets assigned to one and only one material we will additionally need the family of constraints

$$\sum_{j=1}^{M} x_{ij} = 1 \;, \quad \text{for all } i = 1, \ldots, N.$$

Setup Objective Function

Least Squares Objective Function for k^{th} Ray

$$J_k = \frac{1}{v_{\max}^2}\left(L_k - \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_{kj}x_{ij}\right)^2 + \left(T_k - \sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{kj}x_{ij}\right)^2$$

Assuming that the k^{th} ray has ray-parameter p_k, we have defined:

- α_{kj} : Horizontal distance traveled by k^{th} ray with ray-parameter p_k in a layer with velocity v_j.
- β_{kj} : Time spent by k^{th} ray with ray-parameter p_k in a layer with velocity v_j.

Alternating Minimization

Continuous Optimization Problem:

- Hold x_{ij}'s fixed, and minimize over p_k's
- Trivial ray tracing

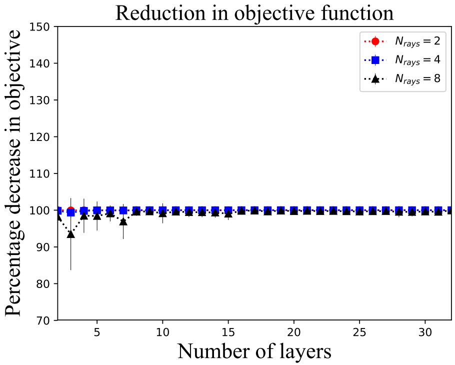
Discrete Optimization Problem:

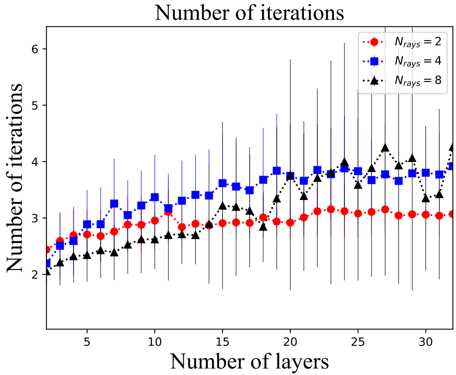
- Hold p_k's fixed, and minimize over x_{ij}'s
- Combinatorial challenge

Classical Computer Trials

- Take total vertical thickness of 1 km.
- Number of materials $M = 3$.
- Materials :
  - Sandstone — 3.0 km / s, Shale — 2.5 km / s, Salt — 4.6 km / s
- Number of sublayers was varied from $N = 2, \ldots, 32$.
- Number of rays was varied from $K = 2, 4, 8$.
- For each combination of N and K, 50 independent problem instances were created. For each instance, uniformly spaced p_k's were used to generate the "true" data.
- Each instance was solved 50 times and statistics were gathered.

Performance of the Algorithm





Pretty good reduction in objective function.

The alternating algorithm converges quite fast in a small number of iterations.

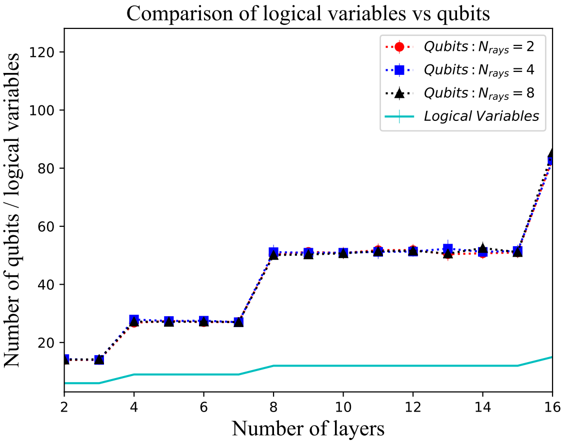
Impact of Number of Rays on Sand-Shale Ratio

- 2 materials, 6 layers
- Different sand-shale ratios
- Vary number of rays: $K = 1, 2, 3, 4, 5$

True Sand Shale Ratio	Observed Sand Shale Ratio				
	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
5.0	5.0	5.0	5.0	5.0	5.0
2.0	2.0	2.0	2.0	2.0	2.0
1.0	1.0	1.0	1.0	1.0	1.0
0.5	0.5	0.5	0.5	0.5	0.5
0.2	0.2	0.2	0.2	0.2	0.2

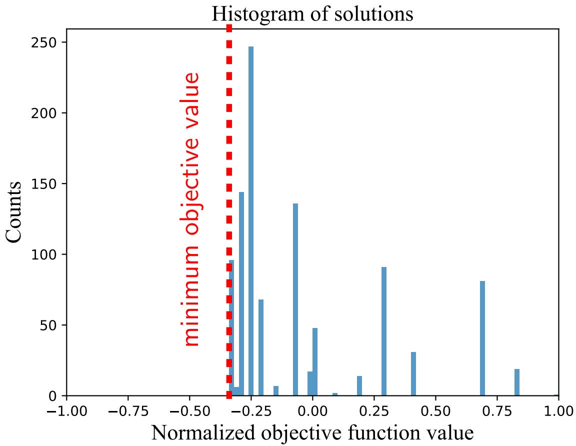
With just 2 materials, we were able to recover the true sand-shale ratios exactly on average. With more materials, increasing the number of rays helps, but our computational experiments suggest that the effect is not monotonic.

How Big Today?



Due to lack of full connectivity of the D-Wave annealer, mapping the problem to the hardware leads to an increase in the number of effective qubits.

Stochasticity of Annealing



*The quantum annealing process has inbuilt stochasticity. Solutions are recovered close to the lowest energy **"ground state"** of the objective function or Hamiltonian.*

Our Thoughts

- Quantum annealer did provide good solutions.
- For number of binary variables beyond 100, solving the problem through enumeration is nearly impossible for a classical computer.
- This method can also be useful for a kind of *"uncertainty quantification"*, as you can potentially generate an ensemble of solutions that satisfy a given error tolerance.

Acknowledgments

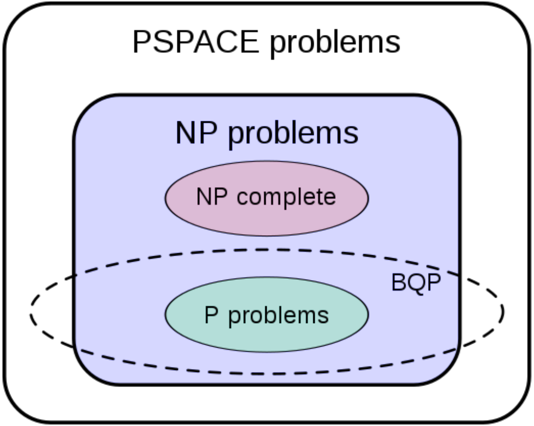
- We thank Peter L. McMahon for many discussions on quantum computing.
- We thank QC Ware Corp. and D-Wave Systems Inc. for giving us permissions to publish the quantum computing results.
- This work was partially supported with funding from the affiliates of the Stanford Exploration Project.

Thank You

Questions?

What Problems Can Quantum Computers Solve Efficiently?

BQP = Bounded-error quantum polynomial time



QUBO to Ising

Model QUBO Problem

$$\begin{aligned} \text{minimize} \quad & x^T Q\, x \\ \text{subject to} \quad & x \in \{0,1\}^n \end{aligned}$$

Conversion to Ising Hamiltonian

Change of variables: $s_i = 2x_i - 1$

$$\begin{aligned} \text{minimize} \quad & \sum_{i>j} J_{ij} s_i s_j + \sum_i h_i s_i \\ \text{subject to} \quad & s \in \{-1,1\}^n \end{aligned}$$

An Optimization Problem With a Lot of Binary Variables

Leads to a non-convex **"mixed integer program"** :

Sum of Squares Optimization Problem

$$
\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{K} J_k \\
\text{subject to} \quad & x_{ij} \in \{0,1\} \;,\; \forall\, i \in \{1,\dots,N\},\; \forall\, j \in \{1,\dots,M\}, \\
& \sum_{j=1}^{M} x_{ij} = 1 \;,\; \forall\, i \in \{1,\dots,N\}.
\end{aligned}
$$

- Continuous variables : The ray-parameters $p_1, \dots, p_K$
- Binary variables : Layer assignments $x_{11}, \dots, x_{NM}$
- These are typically hard problems

Problem Structure

- The continuous problem is separable over each ray! So each J_k can be minimized independently.

$$J_k = \frac{1}{v_{\max}^2}\left(L_k - \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_{kj}x_{ij}\right)^2 + \left(T_k - \sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{kj}x_{ij}\right)^2$$

$$\text{Objective Function :} \quad \sum_{k=1}^{K} J_k$$

- The discrete problem is a *Quadratic Binary Optimization Problem*. Number of binary variables is **NM**.

Question: *Can we reduce the number of binary variables ?*

Towards a Better Combinatorial Optimization Problem: Symmetry of Composition

Symmetry of Composition:

Given an assignment of layers, the objective function is unchanged under arbitrary permutation of layers (this holds because of the horizontal layer assumption). This is a *key observation*.

Key idea: Just count number of times a material repeats. Let y_j denote number of times material j occurs.

- Bound constraint : $0 \leq y_j \leq N$, for all $j = 1, \ldots, M$.
- Sum constraint : $\sum_{j=1}^{M} y_j = N$.

Good but not enough! Only M integer variables, but each variable can take $N + 1$ integer values.

Towards a Better Combinatorial Optimization Problem: Binary Encoding Trick

Binary Encoding Trick:

Since y_j's are positive uniformly spaces integers, work with the binary representation of each variable.

$$y_j = \sum_{l=1}^{r} b_{jl} 2^{l-1} \ , \ \forall \ j = 1, \dots, M \ ,$$

where $r = \lfloor \log_2 N \rfloor + 1$, and $b_{jl} \in \{0, 1\}$.

So now we have $Mr \approx M \log_2 N$ binary variables. This is a $\log \mathbf{N}$ compression.

Equivalent Combinatorial Optimization Problem

Equivalent Optimization Problem

$$
\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{K} \left[\frac{1}{v_{\text{max}}^2} \left(L_k - \sum_{j=1}^{M} \sum_{l=1}^{r} \alpha_{kj} b_{jl} 2^{l-1} \right)^2 \right] + \\
& \sum_{k=1}^{K} \left[\left(T_k - \sum_{j=1}^{M} \sum_{l=1}^{r} \beta_{kj} b_{jl} 2^{l-1} \right)^2 \right] \\
\text{subject to} \quad & b_{jl} \in \{0, 1\} \;\; , \;\; \forall\, j \in \{1, \ldots, M\} \;\; , \;\; \forall\, l \in \{1, \ldots, r\} \; , \\
& \sum_{j=1}^{M} \sum_{l=1}^{r} b_{jl} 2^{l-1} = N.
\end{aligned}
$$

Still a non-convex mixed integer optimization problem.

Alternating Minimization Algorithm

Algorithm 1 Alternating minimization algorithm

procedure ALTERNATING QUBO

// Random assignment
$x_{ij} \leftarrow 0$, for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$
for i = 1 to N **do**
    $j \leftarrow$ Randomly choose from the set $\{1, \ldots, M\}$
    $x_{ij} \leftarrow 1$
Compute $y_j = \sum_{i=1}^{N} x_{ij}$, for all $j = 1, \ldots, M$
Compute b_{jl} as binary representation of y_j, for all $j = 1, \ldots, M$ and $l = 1, \ldots, r$

// Alternating minimization
while Not converged **do**
    $p_k \leftarrow \arg\min_{p_k} J_k$, for all $k = 1, \ldots, K$, and with all b_{jl} fixed.
    $b_{jl} \leftarrow$ solution of QUBO with all p_k fixed.
Compute $y_j = \sum_{l=1}^{r} b_{jl} 2^{l-1}$, for all $j = 1, \ldots, M$

return p_k, y_j for all $k = 1, \ldots, K$ and $j = 1, \ldots, M$

Discrete Search Over Ray-Parameters

Algorithm 2 Global ray parameter search

1: **procedure** DISCRETIZED GLOBAL SEARCH $(k,\ P)$

2: $\Delta p \leftarrow \frac{1}{P\ v_{\max}}$

3: $S \leftarrow \{n\Delta p : n = 0, \ldots, P-1\}$

4: $p_k \leftarrow \underset{p \in S}{\arg\min}\ \frac{1}{v_{\max}^2} \left(L_k - \sum_{j=1}^{M} \frac{\delta p v_j y_j}{\sqrt{1-p^2 v_j^2}} \right)^2 + \left(T_k - \sum_{j=1}^{M} \frac{\delta y_j}{v_j \sqrt{1-p^2 v_j^2}} \right)^2$

5: **return** p_k

P controls level of discretization.

This might seem scary, but is not precisely because ray-parameter search is independent over each ray.

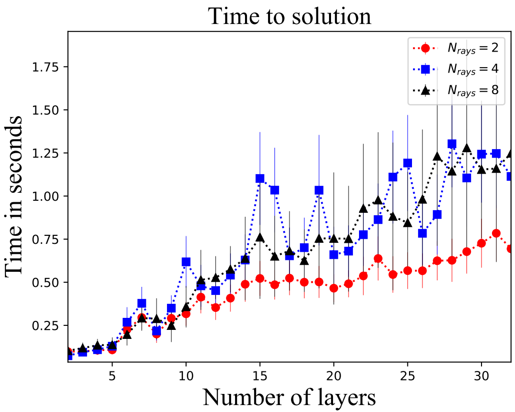
Continuous Search Over Ray-Parameters

Algorithm 3 Newton ray parameter search

1: **procedure** CONTINUOUS LOCAL SEARCH $(k,\ \gamma \in [0,1])$
2: $p_{\min} \leftarrow L_k / (v_{\max}\sqrt{Z^2 + L_k^2})$, $p_{\max} \leftarrow 1/v_{\max}$
3: $p_k \leftarrow \frac{p_{\min}+p_{\max}}{2}$
4: **while** Not converged **do**
5: $p_k \leftarrow p_k - \frac{((1-\gamma)(T(p_k)-T_k)^2+\gamma(L(p_k)-L_k)^2/v_{\max}^2)'}{((1-\gamma)(T(p_k)-T_k)^2+\gamma(L(p_k)-L_k)^2/v_{\max}^2)''}$
6: $p_k \leftarrow \max(p_{\min}, \min(p_{\max}, p_k))$
7: **return** p_k

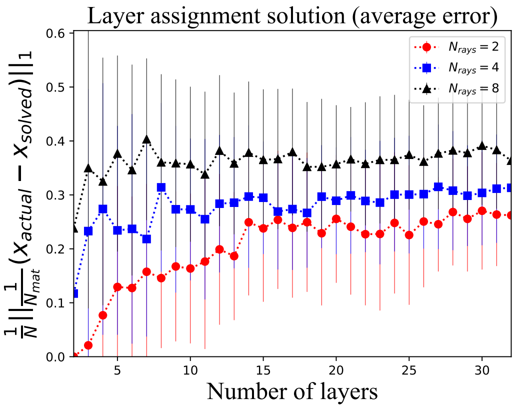
$\gamma \in [0,1]$ controls how travel time and offset terms are weighed in the search process.

Classical Results: Wall Clock Time to Convergence



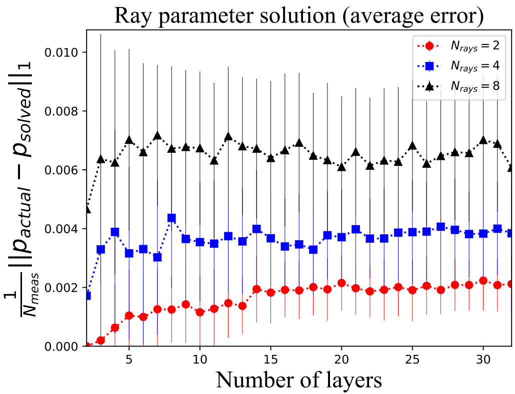
Time to convergence increases with number of layers.

Classical Results: Mean Layer Assignment Error



With the right scaling, errors are constant over number of layers.

Classical Results: Mean Ray-Parameter Error

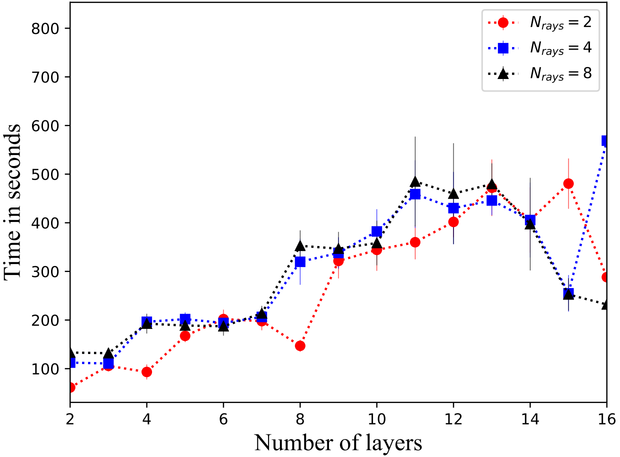


With the right scaling, errors are constant over number of layers.

Preliminary Quantum Solve Trials

- Same experimental setup as in the classical case.
- We varied $N = 2, \dots, 16$, and $K = 2, 4, 8$.
- For each combination of N and K, 10 independent problem instances were created. For each instance, uniformly spaced p_k's were used to generate the "true" data.
- We solved each instance of the QUBO problem on the D-WAVE 2000Q quantum annealer 1000 times and gathered statistics on the total time to solution.

Quantum Annealing Results: Time to Solution



The average time per anneal cycle is about same as the time taken to solve the optimization problem on a classical computer.