Snell tomography using quantum annealing

Stewart A. Levin and Rahul Sarkar

ABSTRACT

By restricting the constituent materials in a horizontally stratified medium to a specific set with well-known acoustic (or elastic) properties, transmission tomography can, in principle, provide the relative fraction of each material within the set of beds over which the rays traverse. In this paper, we formulate an algorithm for this "super-resolution" calculation suitable for quantum computing.

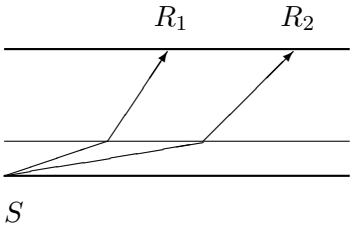
INTRODUCTION

In a horizontally stratified medium, one may arbitrarily permute the layers between source and receiver without changing the recorded traveltime of the direct arrival. This permutation is easily understood because the fixed Snell parameter for any one layer ordering results in a fixed horizontal extent traversal across the layer and a fixed traveltime across the layer. As the horizontal extent of the ray is the sum of the individual horizontal extents of the layers and the total traveltime is the sum of the individual layer traveltimes, any permutation of the layers results only in a permutation of the terms in the sums.

Clearly, a single ray crossing the layers cannot distinguish their arrangement. However, if the velocities of possible layer materials are known or at least tightly bracketed, it may well be possible to determine the fraction of each material in the total package. Of particular interest is net-to-gross or the percentage of sand in, say, a sand-shale package. In that particular two material setting we have two constraints—total traveltime and total horizontal distance traveled by the ray—against which to adjust the fraction of sand versus the fraction of shale. Indeed, for pure vertical propagation time T through a macro-layer thickness Z consisting of materials with velocities v_1 and v_2, the fraction of material 1 is given by

$$\left(\frac{T}{Z} - \frac{1}{v_2}\right) / \left(\frac{1}{v_1} - \frac{1}{v_2}\right) \ .$$

(The case of nonvertical propagation through two materials is worked out in Appendix B.) Similarly, with N distinct constant velocity materials and a corresponding number of at least $N-1$ independent rays there is no a priori reason not to expect a unique set of fractions for a larger number of material compositions.



In the setting of transmission surveys there are often additional measurements to match that span various ranges of layers. This theoretically reduces uncertainty in the calculations, albeit at the cost of a combinatoric explosion of possible layer fractions to resolve. Quantum computing holds the promise of exploring many combinations of parameters at the same time. A naive example would be to finely divide each layer into many minilayers and calculate the traveltime misfit for every possible choice of distinct material for each layer.

ALGORITHMS

As the ordering of fine layering is immaterial to calculating traveltimes, the first simplification is to assume materials are ordered from slowest to fastest within each layer being subdivided. A continuous variable formulation for a layer of thickness Z posits unknown sublayer thicknesses δ_i, $i = 1 \dots N$ that total Z and seeks the constant Snell ray parameter p that allows a ray to reach from source to receiver and match the known traveltime T. Of course, measurement errors mean we may not be able to precisely match the known endpoint at the known traveltime; therefore we formulate the problem as a least-square fit of both horizontal deviation and traveltime deviation for a given trial Snell parameter and sublayer thicknesses. To balance time misfit against distance misfit, we choose to divide horizontal misfit by the fastest layer velocity.

Let the height of the macrolayer be Z and its width L. We have, as above, $Z = \sum_{i=1}^{N} \delta_i$. Denoting the material velocities of the N layers by v_i, Snell's law (Slotnick and Geyer, 1959) states that for an individual sublayer i,

$$pv_i = \sin\theta_i \ . \tag{1}$$

Therefore,

$$\tan\theta_i = \frac{pv_i}{\sqrt{1 - p^2 v_i^2}} \tag{2}$$

and the horizontal traversal of a ray through the i^{th} sublayer is given by

$$l_i = \frac{p\delta_i v_i}{\sqrt{1 - p^2 v_i^2}} \ . \tag{3}$$

Finally, the traveltime through the sublayer is

$$t_i = \frac{\delta_i}{v_i\sqrt{1 - p^2 v_i^2}} \ . \tag{4}$$

Our least-squares formulation asks that, given K traveltime-offset measurements, we minimize

$$J = \sum_{\text{rays}} \frac{1}{2} \left[\frac{1}{v_{\max}^2} (L^{\text{ray}} - \sum_{i=1}^{N} l_i^{\text{ray}})^2 + (T^{\text{ray}} - \sum_{i=1}^{N} t_i^{\text{ray}})^2 \right] \tag{5}$$

over p values for the K rays and the set of thicknesses δ_i. Of course the δ_i's must not be negative.

Before we continue, there are two things to note. First, using reciprocity as needed, we can take p to be nonnegative, e.g., traversing left to right; and therefore, limit the value of p between 0 and $1/v_{\max}$. The lower bound of 0 can be improved by noting that the straight line between source and receiver is the smallest value possible with the fastest material velocity, and so, $p_{\min} = L/(v_{\max}\sqrt{Z^2 + L^2})$. Second, there is a possibility that some sublayer thickness should be zero. To handle this possibility, we need to also progressively drop the fastest sublayer velocity and recalculate with the new sub-stack and new p upper and lower bounds.

The δ_i occur linearly in the l_i and t_i variables; and therefore, for a fixed guess for p, the minimization leads to a linear problem, quite suited for a conventional computer. For our choice of continuous algorithm, we solve for the δ_i's using the MATLAB® *lsqlin* approach, alternating between that and searching over possible p's.

Searching over p is not as simple as it might seem, which is because ray tracing can be discontinuous as a layer shrinks to zero thickness. For example, consider a two layer model with composite dimensions 1 km vertically and 2 km horizontally and ask for the ray that connects the top left to the bottom right. Let the two materials have velocities 1 km/s and 2 km/s respectively. If the low velocity layer has zero thickness, $p = 1/\sqrt{5} = 0.4472136$. If the high velocity layer has zero thickness, $p = 0.5/\sqrt{5} = 0.223607$. Calculating for intermediate fractions of lower-velocity material yields the following table:

Z	p
0.00	0.4472136
0.25	0.4640339
0.50	0.4802597
0.75	0.4938107
0.99	0.4999877
1.00	0.2236068

The discontinuity arises because the equations permit near horizontal propagation in the fastest velocity. This discontinuity emphasizes why we include an outer iteration that drops materials one at a time from fastest to slowest.

However, Levin (2012) has noted that using Newton's method to determine p for two point ray tracing to match a given offset in this setting is globally convergent because the first and second derivatives with respect to p,

$$\frac{dl_i}{dp} = \frac{\delta_i v_i}{(1 - p^2 v_i^2)^{3/2}} \quad , \quad \frac{d^2 l_i}{dp^2} = \frac{3\delta_i p v_i^3}{(1 - p^2 v_i^2)^{5/2}} \quad ,$$

are positive in $(0, 1/v_{\max})$, which is also true for matching a given arrival time as

$$\frac{dt_i}{dp} = \frac{\delta_i p v_i}{(1-p^2v_i^2)^{3/2}} \ , \quad \frac{d^2t_i}{dp^2} = \frac{\delta_i v_i (1+2pv_i^2)}{(1-p^2v_i^2)^{5/2}} \ .$$

This equation suggests that we might be quite effective in approximating the effect of changing p with a quadratic derived from an initial ray and a handful[1] of Newton iterations. In particular, for a given set of layering δ_i and a given k, the minimum of equation 5 lies between the ps that separately minimize the offset and arrival time terms. Our numerical experience with the three material, single macrolayer case has been that initializing p to its minimum feasible value and Newton iterating only over the traveltime fit has so far led us to a global minimum.

COMBINATORIAL OPTIMIZATION

In this section, we leverage ideas from the previous section to formulate a combinatorial optimization problem that in principle seeks to minimize the same quantity J, as in equation 5. Let us assume that all the N sublayer thicknesses are equal, which we denote as δ; and therefore, we have $\delta = Z/N$. Let $\mathbf{V}$ be the finite set of M material velocities. The velocity of the j^{th} material is denoted as v_j, and we assume that all the v_js are unique. We introduce binary decision variables $x_{ij} \in \{0, 1\}$ to represent the assignment of sublayer i to material j, and define as follows:

$$x_{ij} = \begin{cases} 1 & \text{if sublayer } i \text{ is material } j \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$. To ensure that each sublayer gets assigned to one and only one material we will additionally need the family of constraints

$$\sum_{j=1}^{M} x_{ij} = 1 \ , \ \text{ for all } i = 1, \ldots, N. \tag{7}$$

Assuming that we have a total of $K \geq M-1$ rays that can be used in the fitting process, we denote the ray parameter for the k^{th} ray by p_k, the horizontal traversal of the ray through the i^{th} sublayer by l_{ki}, the traveltime through the i^{th} sublayer by t_{ki}, the known total horizontal traversal by L_k, and the known traveltime by T_k. Then, analogous to formulas (3) and (4), we have

$$l_{ki} = \sum_{j=1}^{M} \alpha_{kj} x_{ij} \ , \ \ t_{ki} = \sum_{j=1}^{M} \beta_{kj} x_{ij} \ , \tag{8}$$

where

$$\alpha_{kj} = \frac{\delta p_k v_j}{\sqrt{1-p_k^2 v_j^2}} \ , \ \ \beta_{kj} = \frac{\delta}{v_j \sqrt{1-p_k^2 v_j^2}} \ . \tag{9}$$

[1]While globally convergent, the first few Newton iterations may swing around a fair amount and only converge quadratically in the later iterations.

Our misfit function J_k for the k^{th} ray is

$$J_k = \frac{1}{v_{\max}^2}\left(L_k - \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_{kj}x_{ij}\right)^2 + \left(T_k - \sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{kj}x_{ij}\right)^2 , \tag{10}$$

and we seek to minimize the sum of these misfit functions over all the K rays, which is the quantity $J = \sum_{k=1}^{K} J_k$, where the minimization takes place over all the binary variables x_{ij} and all the ray parameters p_k, thereby leading to the optimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{k=1}^{K} J_k \\ &\text{subject to} \quad x_{ij} \in \{0,1\} \ , \ \forall\, i \in \{1,\ldots,N\}, \ \forall\, j \in \{1,\ldots,M\}, \\ &\qquad\qquad \sum_{j=1}^{M} x_{ij} = 1 \ , \ \forall\, i \in \{1,\ldots,N\}. \end{aligned} \tag{11}$$

Invariance under symmetry of composition

The number of binary variables involved in the optimization problem in equation 11 is NM. For typical applications, the number of materials, M, is a small number and is fixed for a given problem which we will think of as a constant, but we want to be able to study the behavior of solutions as we refine our layer discretization δ, by increasing N. Clearly it raises the question of whether or not we can reduce the dependency of the number of binary variables needed to encode the problem. As we show next, this reduction is indeed possible by exploiting a symmetry of the objective function.

We start by noticing that we can interchange the order of summation over i and j in equation 10, and introducing the integer variables $y_j = \sum_{i=1}^{N} x_{ij}$ for all $j = 1,\ldots,M$, we have

$$\begin{aligned} J_k &= \frac{1}{v_{\max}^2}\left(L_k - \sum_{j=1}^{M}\alpha_{kj}\sum_{i=1}^{N}x_{ij}\right)^2 + \left(T_k - \sum_{j=1}^{M}\beta_{kj}\sum_{i=1}^{N}x_{ij}\right)^2 \\ &= \frac{1}{v_{\max}^2}\left(L_k - \sum_{j=1}^{M}\alpha_{kj}y_j\right)^2 + \left(T_k - \sum_{j=1}^{M}\beta_{kj}y_j\right)^2 . \end{aligned} \tag{12}$$

This equation suggests an important symmetry, namely the *"symmetry of composition,"* which states that the quantity J_k is invariant with respect to permutations of the sublayers as long as the number of sublayers of each material type is kept constant. Moreover, this is true for each k, which means that it is true for the quantity $J = \sum_{k=1}^{K} J_k$. Such permutations also do not lead to any constraint violations in equation 11 as it simply involves relabeling of the binary variables. Thus, any feasible solution of equation 11, not necessarily optimal, has the property that permuting the sublayers leads to another feasible solution with exactly the same objective function value.

It is possible to remove this redundancy by formulating an optimization problem that is equivalent to equation 11, in terms of the new variables y_j. Transformation of the objective

function has already been carried out in equation 12; and therefore, we only need to worry about the constraints, which follow immediately by noticing that each integer variable y_j is confined to the range $0 \leq y_j \leq N$ because

$$\sum_{j=1}^{M} y_j = \sum_{j=1}^{M}\sum_{i=1}^{N} x_{ij} = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij} = \sum_{i=1}^{N} 1 = N \ . \tag{13}$$

This observation leads to the equivalent optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K}\left[\frac{1}{v_{\max}^2}\left(L_k - \sum_{j=1}^{M}\alpha_{kj}y_j\right)^2 + \left(T_k - \sum_{j=1}^{M}\beta_{kj}y_j\right)^2\right] \\ \text{subject to} \quad & y_j \in \{0,\ldots,N\} \ , \ \forall\, j \in \{1,\ldots,M\}, \\ & \sum_{j=1}^{M} y_j = N \ . \end{aligned} \tag{14}$$

Notice that once we have y_j for all j, any assignment of sublayers to materials that satisfy $\sum_{i=1}^{N} x_{ij} = y_j$ will be a valid assignment. For example, such an assignment can be trivially generated by assigning the first y_1 sublayers to material 1, the next y_2 sublayers to material 2, etc.

Binary encoding trick

Although the optimization problem in equation 14 is an improvement over equation 11 in terms of the number of variables, we now have integer variables that are no longer binary and instead take values from the set $S = \{0,\ldots,N\}$. However, because the elements of S are uniformly spaced, an additional *"binary encoding"* trick can be employed that converts the optimization problem in equation 14 into one involving binary variables, at the expense of increasing the number of variables by a factor of roughly $\log_2 N$. The binary encoding trick involves writing each y_j in its base-2 representation. For each j, we introduce r binary variables $b_{j1},\ldots,b_{jr}$, such that

$$y_j = \sum_{l=1}^{r} b_{jl}2^{l-1} \ , \ \forall\, j = 1,\ldots,M \ , \tag{15}$$

where $r = \lfloor \log_2 N \rfloor + 1$. Direct substitution into equation 14 gives the equivalent optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K}\left[\frac{1}{v_{\max}^2}\left(L_k - \sum_{j=1}^{M}\sum_{l=1}^{r}\alpha_{kj}b_{jl}2^{l-1}\right)^2 + \left(T_k - \sum_{j=1}^{M}\sum_{l=1}^{r}\beta_{kj}b_{jl}2^{l-1}\right)^2\right] \\ \text{subject to} \quad & b_{jl} \in \{0,1\} \ , \ \forall\, j \in \{1,\ldots,M\} \ , \ \forall\, l \in \{1,\ldots,r\} \ , \\ & \sum_{j=1}^{M}\sum_{l=1}^{r} b_{jl}2^{l-1} = N. \end{aligned} \tag{16}$$

The optimization problem in equation 16 is still over all the ray parameters p_k and all the binary variables b_{jl}, but the important difference is that compared to the optimization problem in equation 11, which required MN binary variables; we now only require Mr binary variables, which is approximately $M \log_2 N$, a logarithmic improvement in N.

Alternating minimization

The optimization problem in equation 16 is a mixed-integer non-convex optimization problem and is difficult to solve in general. However, if we fix the binary decision variables b_{jl} subject to the constraints, then the objective function becomes separable in each ray, indexed by the variable k, and the resulting minimization problem for fixed k only involves searching over a single ray parameter p_k. This can be done using one of the two methods described in Appendix A.

Otherwise, once we fix the ray parameters for all K rays, the objective function in equation 16 becomes a quadratic function of the binary variables b_{jl}, subject to the linear constraints. This problem can be written in an equivalent Quadratic Unconstrained Binary Optimization (QUBO) form, as outlined in Appendix C, which can then be solved using a quantum annealer. These observations motivate us to formulate a heuristic for solving equation 16 using an alternating minimization strategy, outlined in algorithm 1.

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 $\{1, \ldots, M\}$

$x_{ij} \leftarrow 1$

end for

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 in (16) with all p_k fixed.

end while

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 $j = 1, \ldots, M$

end procedure

For comparison, we use algorithm 2 in our numerical experiments to seek a continuous variable solution on a conventional computer using convex optimization and Newton's method for traveltime fit and iterates over the set of compositions of N minilayers to avoid getting trapped in local minima.

In the general case of inverting N_{ray} arrivals from N_{macro} macro-layers divided individually into N_{mini} minilayers consisting of up to M materials, the complexity of algorithm 2 is

$$O\left(N_{\text{ray}} \cdot \binom{M + N_{\text{mini}} - 1}{M - 1}^{N_{\text{macro}}}\right) .$$

Algorithm 2 Continuous minimization with Newton refinement algorithm

procedure ALTERNATING QUBO WITH NEWTON REFINEMENT

    // Iterate over different number of materials
    for $\mu = M$ to 1 materials, ordered fastest to slowest **do**

        // Iterate over all compositions
        for Each composition of N layers into μ materials **do**

            // Assign layer thicknesses
            $y_j \leftarrow$ Assign based on composition, for all $j = 1, \ldots, \mu$

            // Iterate over the rays and perform Newton refinement
            for $k = 1$ to K rays **do**
                while Not converged **do**
                    $p_k \leftarrow p_k - (T(p_k) - T_k)/T'(p_k)$, with all y_j fixed
                end while
            end for

            // Relax layer thicknesses to take continuous values
            $\tilde{y}_j \leftarrow$ solution of constrained convex optimization with all p_k fixed

            Record $(p_k, \mu, \tilde{y}_j)$ pairs with the best fit, for all $k = 1, \ldots, K$ and $j = 1, \ldots, \mu$
        end for
    end for

    return p_k, μ, $\tilde{y}_j$ for all $k = 1, \ldots, K$ and $j = 1, \ldots, \mu$
end procedure

NUMERICAL EXPERIMENTS

In this section we provide some numerical results obtained by running algorithms 1 and 2 on specific instances of the tomography problem. We first present a detailed study of the full alternating algorithm to understand how it is performing in practice on different problem sizes, and how efficient it is at solving these problem instances. We then present some preliminary results obtained by solving a few small problem instances of only the QUBO subproblem on a quantum annealer.

Analysis of the overall algorithm performance

The first aspect we report on is the performance of algorithm 1 with respect to finding a good solution. This was conducted by solving both steps of the process using a classical CPU based computer. In particular, the QUBO problems were solved using the CPLEX® optimization library. The problem sizes were kept small for each QUBO instance to be solvable in a reasonable amount of time, allowing us to generate statistics over a large number of runs. It is to be noted that as the problem sizes grow, solving for the layer assignments becomes increasingly more difficult using classical search heuristics, such as those employed by CPLEX, because of the combinatorial nature of the problem.

For our experiments, we set up a single macro-layer of thickness 1 km, and three constituent materials—Sandstone, Shale and Salt—with velocities given by 3.0 km/s, 2.5 km/s, and 4.6 km/s, respectively. We considered different experimental geometries by varying the number of minilayers N from 2 to 32; and similarly, we also varied the number of measurements N_{meas} to take values 2, 4, and 8. For each value of N_{meas}, the ray parameters were chosen by uniformly dividing the interval $[0, 1/v_{\max})$ into that many points. Next, we randomly assigned the minilayers to create our ground truth model, and traced rays through it to generate the L_j and the T_j. This gave us one instance of our optimization problem, and we generated 50 such instances for each value of N and N_{meas}. We solved each instance 50 times using algorithm 1 to gather meaningful statistics. In all the cases, the average reduction in the objective function was close to 100%, thereby indicating that the alternating algorithm is performing well at minimizing the objective function. We should also note that algorithm 1 guarantees that the objective function reduction is monotonic if the QUBOs are solved exactly, which was also confirmed in our experiments.

In Figure 1, we show several key performance indicators of the convergence properties of the algorithm, such as the number of iterations and time taken until convergence in a) and b), respectively, and the errors in the solved variables as compared to the actual ones in c) and d). These plots show some obvious facts such as an increasing trend in the runtime and the number of iterations as the number of variables increase, but it also exhibits some phenomena that are harder to explain—for example, while the errors in the ray parameters appear to stay flat as we increase the number of minilayers, the errors in the layer assignments show a linearly increasing trend.

For algorithm 2, our testing was run on a 2-GHz central processing unit (CPU). Running a single case with the same materials and all 5151 compositions of 100 minilayers took 7 minutes and 28 seconds, working out to an average of a shade less than one tenth of a second per composition.

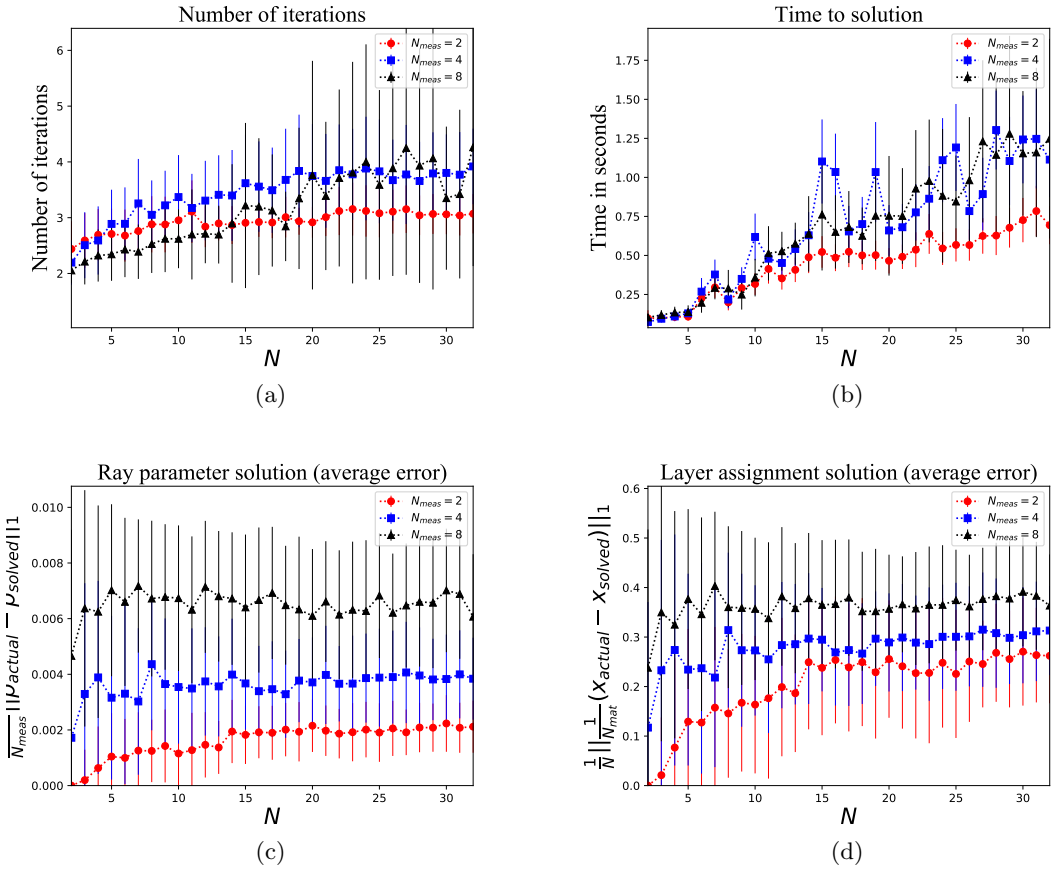


Figure 1: Plots of various performance metrics for algorithm 1, averaged over runs with the error bars representing the standard deviations of the quantities. The horizontal axis in all the plots represent the number of minilayers N. The quantities plotted clockwise from top-left are: a) number of iterations to convergence, b) wall clock time to convergence, c) mean l_1 error in the ray-parameter solution, and d) mean l_1 error in the layer assignments. [**CR**]

Initial quantum annealing results

Here we present and examine our results of solving the QUBO sub-problem on the D-Wave 2000Q quantum annealer. We consider the same experimental setup, as in the last section. In the current annealers, a practical limitation affecting performance is the lack of hardware coupling between all the qubits. Therefore, in order to embed a general QUBO with fully connected topology, copies of the qubits are necessary. We refer to the total qubits after embedding as the "qubit count," and the number of binary variables in the original QUBO as the "logical variable count.". In our problem, once we fix the number of materials, the logical variable count grows logarithmically in N, and is independent of N_{meas}. The resulting qubit counts are shown in Figure 2. We observe that N_{meas} has almost no impact on the qubit count, as we can see from the plot that the variance is near zero.

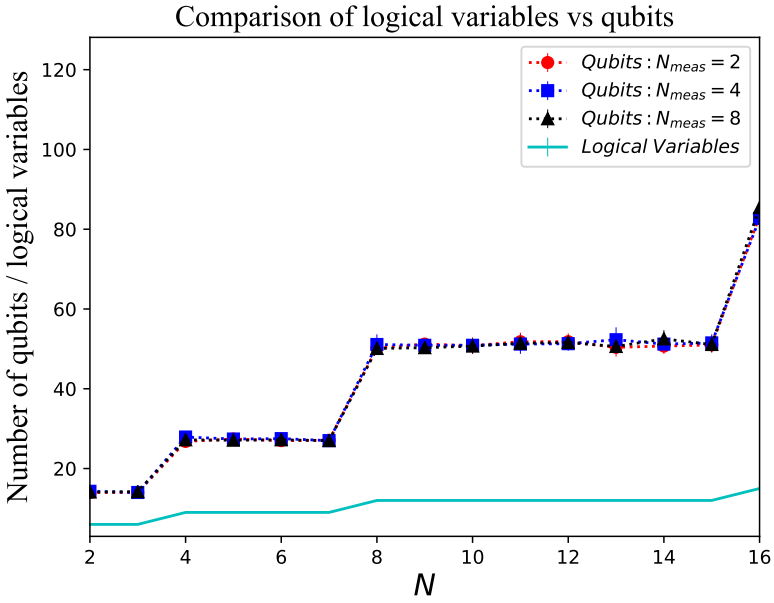


Figure 2: Mean counts for qubits plotted with their standard deviations for different values of N and N_{meas} for three materials. The logical variable count has also been plotted. [**CR**]

Another important performance metric is the time taken to solution. We plot this metric for values of N ranging from 2 to 16, and N_{meas} taking values 2, 4, 8 in Figure 3. For each parameter combination we generated 10 QUBO problem instances with random initializations, and solved each problem 1000 times on the annealer. The time plotted is the total time to solution that involves network communication costs, time needed to embed the problem on the annealer, etc., in addition to the actual anneal time for all the 1000 runs. As expected, the time increases with the problem size N; although interestingly even with overhead costs, the time taken per anneal cycle has a maximum value of approximately 0.5 s among the problem sizes tried, which is quite comparable to the overall runtime of the alternating algorithm on classical computers, for similar problem sizes.

Adiabatic quantum computation by its very nature has stochasticity built into it—that is, the annealing process gives us a solution close to the ground state of the QUBO

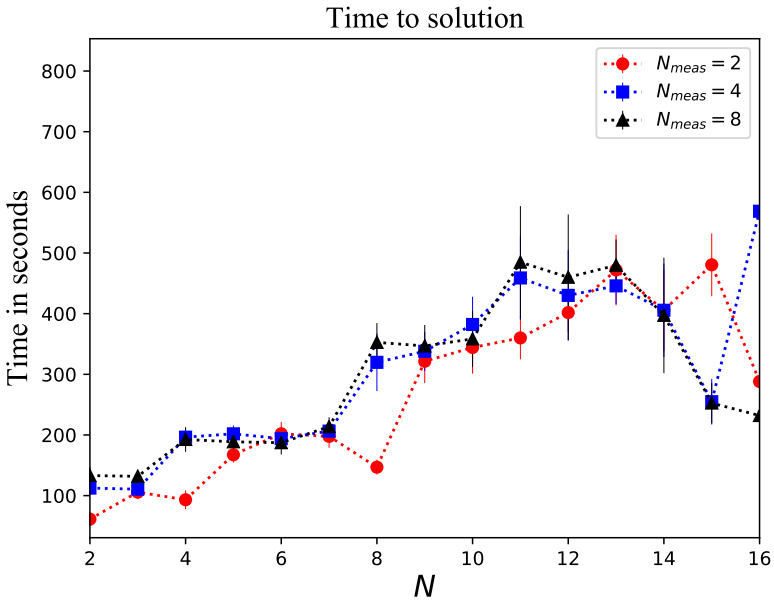


Figure 3: Total time to solution for 1000 anneal cycles plotted with their standard deviations for different values of N and N_{meas} for three materials. [**NR**]

objective, but not always exactly the ground state. This process is demonstrated for a particular QUBO instance in Figure 4, where we have plotted a histogram of the QUBO objective function (which can be negative because we drop constant terms in the QUBO formulation) corresponding to the solutions obtained for 1000 repeated annealing trials. As we can see, most of the solutions are clustered around the global minimum.

CONCLUSIONS AND FUTURE WORK

We formulated our Snell tomography problem for laterally homogenous media and a finite set of material velocities as a mixed integer problem and proposed an alternating algorithm to solve it. Our numerical studies suggest that the algorithm performs reasonably well at solving the optimization problem. Although we can only guarantee convergence to a local minimum, the objective function reduction is on an average over 99%, which is good for most practical applications. We studied the solution of the QUBO sub-problem on a quantum annealer to obtain the qubit counts after the embedding process, and get an estimate of the runtimes, and found them to be reasonable for different problem sizes.

For future work on the quantum annealing aspect, a detailed suite of performance benchmarking tests needs to be performed to evaluate important metrics such as the accuracy of the solutions as compared to the ground state, statistics of the solutions with respect to constraint satisfaction, etc. Another very interesting aspect that remains to be studied is the impact of quantum solves for the QUBO subproblems on the complete alternating algorithm. The inherent stochasticity of quantum annealing may allow the alternating algorithm to escape local minima, and be an aspect that we study in the future. Finally, we are exploring possible ways the full Snell tomography problem, that is including a search

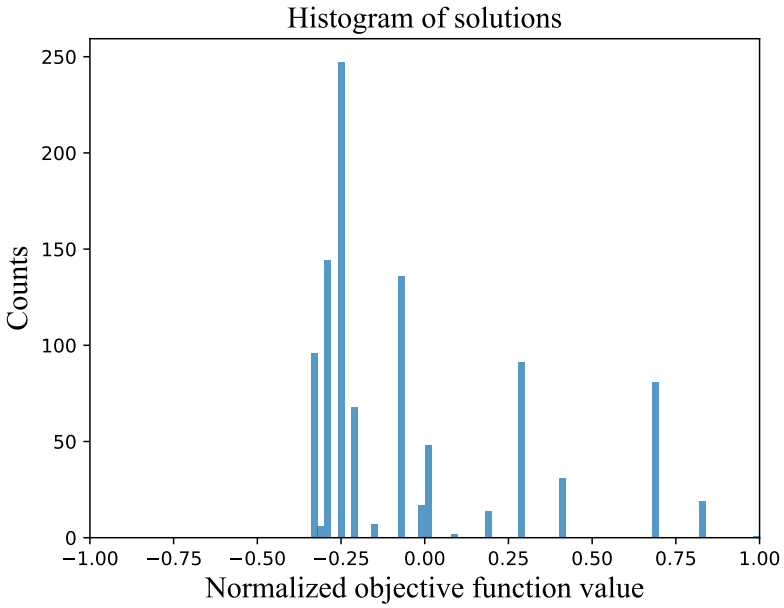


Figure 4: Histogram of objective function values (normalized by the maximum value obtained) corresponding to solutions for 1000 anneal cycles for a single QUBO instance. [**NR**]

over a discrete set of ray parameters such as developed in Appendix D, might feasibly be run on the current generation of quantum annealers.

ACKNOWLEDGMENTS

The first author would like to thank Peter L. McMahon for proof-reading sections of this manuscript, and for his useful suggestions on the quantum annealing metrics studied in this paper. We would like to thank QC Ware, Corp. whose software platform was used to run the quantum annealing experiments; and D-Wave Systems Inc. whose quantum annealing hardware the experiments were run on. We also thank both QC Ware, Corp. and D-Wave Systems Inc. for giving us permissions to publish this work.

REFERENCES

Gauss, C. F., 1828, Beweis eines algebraischen lehrsatzes: Crelle's Journal für die reine und angewandte Mathematik, **3**, 1–4. (Translation by Stewart A. Levin at http://sepwww.stanford.edu/data/media/public/oldsep/stew/gausstrn.pdf).

Hansen, P., 1979, Methods of nonlinear 0-1 programming: Annals of Discrete Mathematics, **5**, 53–70.

Levin, S. A., 2012, Two point raytracing for reflection off a 3D plane: SEP Report, **148**, 89–100.

Slotnick, M. M. and R. A. Geyer, 1959, Lessons in seismic computing: Society of Exploration Geophysicists, Tulsa, Oklahoma.

APPENDIX A

In this appendix, we present two algorithms that can be used to perform the search over the ray parameter p_k for the k^{th} ray given that the layer assignments are fixed, or alternatively when the binary variables b_{jl} in equation 16 are fixed for all $j = 1, \ldots, M$ and $l = 1, \ldots, r$. The first algorithm takes a discrete sampling approach to the problem to locate the global minima of this one dimensional search problem up to discretization errors. The second algorithm is the descent based Newton's algorithm that converges to a local minima.

Global search based on discretization

The optimization problem that we are solving for a fixed ray k and fixed layer assignments is given by the following:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{v_{\max}^2}\left(L_k - \sum_{j=1}^{M} \frac{\delta p_k v_j y_j}{\sqrt{1-p_k^2 v_j^2}}\right)^2 + \left(T_k - \sum_{j=1}^{M} \frac{\delta y_j}{v_j\sqrt{1-p_k^2 v_j^2}}\right)^2 \\ &\text{subject to} \quad p_k \in [0, 1/v_{\max}) \end{aligned} \tag{A-1}$$

where we have substituted the expressions for α_{kj}, β_{kj} and y_j from equation 9 and equation 15 in the objective function of equation 16 for a single ray k. We discretize the domain $[0, 1/v_{\max})$, into P uniformly spaced values for some positive integer P, and choose the value that attains the minimum objective function. The pseudocode is outlined in algorithm A-1, which takes as input the index k of the ray and the discretization parameter P.

Algorithm A-1 Global ray parameter search

procedure DISCRETIZED GLOBAL SEARCH (k, P)
  $\Delta p \leftarrow \frac{1}{P\, v_{\max}}$
  $S \leftarrow \{n\Delta p : n = 0, \ldots, P-1\}$
  $p_k \leftarrow \arg\min_{p \in S} \frac{1}{v_{\max}^2}\left(L_k - \sum_{j=1}^{M} \frac{\delta p v_j y_j}{\sqrt{1-p^2 v_j^2}}\right)^2 + \left(T_k - \sum_{j=1}^{M} \frac{\delta y_j}{v_j\sqrt{1-p^2 v_j^2}}\right)^2$
  return p_k
end procedure

Newton's method

In view of the global convergence of Newton's method for ray tracing in a horizontally isotropic layered medium (Levin, 2012), for fixed layer assignment y_j, the ray parameters p_k can be updated using algorithm A-2. It is easily seen that this is just Newton's root finding algorithm for a stationary point of the objective function in (A-1), with a slight modification that forms a convex combination of the traveltime and offset misfit terms. In this case we are only guaranteed to converge to a local minima, as the global convergence guarantee does not hold any more with both the offset and traveltime misfit terms present. We finally note that we used $\gamma = 0$ in our numerical experiments as that option appeared to avoid trapping us in a local minima of the larger optimization problem.

Algorithm A-2 Newton ray parameter search

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

APPENDIX B

Two material analytic formula

As noted in this report, material percentages for a two-layer system are resolvable with one offset-time pair and a vertical ray example was shown. For an offset ray, an analytic formulation is significantly more complex and results in a sixth order polynomial to solve. Our derivation follows.

Let the ray traverse from $(0,0)$ to (L,Z) in time T, and let A and B be the vertical division of Z into between the velocities v_1 and v_2. We then have four equations to combine:

$$Z = A + B \quad \text{vertical distance,} \tag{B-1}$$

$$L = A\tan\theta_1 + B\tan\theta_2 \quad \text{horizontal distance,} \tag{B-2}$$

$$T = \frac{A}{v_1\cos\theta_1} + \frac{B}{v_2\cos\theta_2} \quad \text{traveltime, and} \tag{B-3}$$

$$\frac{v_1}{v_2} = \gamma = \frac{\sin\theta_1}{\sin\theta_2} \quad \text{Snell's Law.} \tag{B-4}$$

Eliminating $B = Z - A$ yields

$$A\left(\tan\theta_1 - \tan\theta_2\right) = L - Z\tan\theta_2 \tag{B-5}$$

$$A\left(\frac{1}{v_1\cos\theta_1} - \frac{1}{v_2\cos\theta_2}\right) = T - \frac{Z}{v_2\cos\theta_2} \quad . \tag{B-6}$$

Eliminating A

$$\frac{L - Z\tan\theta_2}{\tan\theta_1 - \tan\theta_2} = \frac{T - \frac{1}{v_2\cos\theta_2}Z}{\frac{1}{v_1\cos\theta_1} - \frac{1}{v_2\cos\theta_2}} = \frac{Tv_2\cos\theta_2 - Z}{\frac{v_2\cos\theta_2}{v_1\cos\theta_1} - 1} \tag{B-7}$$

and using Snell's Law B-4 to eliminate θ_1 gives us

$$\frac{L - Z\tan\theta_2}{\gamma\sin\theta_2/\sqrt{1-\gamma^2\sin^2\theta_2} - \tan\theta_2} = \frac{Tv_1\cos\theta_2 - \gamma Z}{\cos\theta_2/\sqrt{1-\gamma^2\sin^2\theta_2} - \gamma} \tag{B-8}$$

to solve for $\sin\theta_2$; and therefore, p. Denoting $s = \sin\theta_2$ and replacing $\cos\theta_2$ with $\sqrt{1-s^2}$, we expand

$$\frac{L - Zs/\sqrt{1-s^2}}{\gamma s/\sqrt{1-\gamma^2 s^2} - s/\sqrt{1-s^2}} = \frac{Tv_1\sqrt{1-s^2} - \gamma Z}{\sqrt{1-s^2}/\sqrt{1-\gamma^2 s^2} - \gamma} \quad , \tag{B-9}$$

multiply by $s/\sqrt{1-\gamma^2 s^2}$

$$\frac{L - Zs/\sqrt{1-s^2}}{\gamma - \sqrt{1-\gamma^2 s^2}/\sqrt{1-s^2}} = s\frac{Tv_1\sqrt{1-s^2} - \gamma Z}{\sqrt{1-s^2} - \gamma\sqrt{1-\gamma^2 s^2}} \quad , \tag{B-10}$$

and simplify to obtain

$$\frac{L\sqrt{1-s^2} - Zs}{\gamma\sqrt{1-s^2} - \sqrt{1-\gamma^2 s^2}} = s\frac{Tv_1\sqrt{1-s^2} - \gamma Z}{\sqrt{1-s^2} - \gamma\sqrt{1-\gamma^2 s^2}} \quad . \tag{B-11}$$

Subtracting the right side from the left and simplifying yields

$$0 = \frac{(Tv_1 s - \gamma L)\sqrt{1-\gamma^2 s^2} - (1-\gamma^2)Zs - (\gamma Tv_1 s - L)\sqrt{1-s^2}}{\left[(\gamma^2+1)(1-s^2)\sqrt{1-\gamma^2 s^2} + \gamma((\gamma^2+1)s^2 - 2\right]\sqrt{1-s^2}} \quad . \tag{B-12}$$

Before continuing, we note that both the numerator and denominator vanish when $\gamma = 1$. This is reassuring because if the two layer velocities are the same, we can choose any boundary or boundaries between them and not change the offset and traveltime of the (straight) ray.

Setting the numerator equal to zero, we eliminate the radicals by first moving the term with $\sqrt{1-\gamma^2 s^2}$ to one side, squaring both sides, rearranging to put $\sqrt{1-s^2}$ to one side and squaring again to generate a nominally eighth order polynomial. However, the 8^{th}, 7^{th}, and 1^{st} order terms cancel, leaving the 6^{th} order polynomial equation

$$\begin{aligned} 0 \;=\; & 4\gamma^2(\gamma^2-1)^2 T^2 v_1^2 (Z^2+L^2)\, s^6 \\ & -4\gamma(\gamma^2-1)^2 Tv_1 L[(\gamma^2+1)(L^2+Z^2) + T^2 v_1^2]\, s^5 \\ & +(\gamma^2-1)^2[(\gamma^2-1)^2 Z^4 + 2(\gamma^4+1)L^2 Z^2 + (\gamma^2+1)^2 L^4 \\ & \qquad +2(\gamma^2+1)T^2 v_1^2 (L^2 - Z^2) + T^4 v_1^4]\, s^4 \\ & +4\gamma(\gamma^2-1)^2 Tv_1 L(L^2+2Z^2)\, s^3 \\ & -2(\gamma^2-1)^2 L^2[(\gamma^2+1)(L^2+Z^2) + T^2 v_1^2]\, s^2 \\ & +(\gamma^2-1)^2 L^4 \end{aligned} \tag{B-13}$$

in $s = \sin\theta_2$.

Regardless of the sign of the s^4 term, there are four sign changes in the coefficients. Applying Descartes' Rule of Signs (Gauss, 1828), we conclude that there are either zero, two or four positive roots. One may further limit the number of roots between 0 and 1 using the change of variable $u = 1/s - 1$, which is the same as recasting B-13 in terms of $\tan\frac{\theta_2}{2}$. For the test case ($v_1 = 1.7, v_2 = 2.2, Z = 1, L = 2, T = 1.061728621$), there are two sign changes in the coefficients of the resulting 6^{th} order u polynomial. One root has $\sin\theta_2 = 0.913856216$, corresponding to the ray actually traced for the test with $A = 0.2$ and $B = 0.8$. The other root, $\sin\theta_2 = 0.874985873$, corresponds to a negative value of $\cos\theta_2$ and a negative value of A. This nonphysical solution is understandable because of the repeated squaring we did to obtain equation B-13 erased signs.

APPENDIX C

Transformation to QUBO form

We carry out the reduction of the constrained binary optimization problem in equation 16 to a quadratic unconstrained binary optimization problem (QUBO), using a well-known transformation process. Let $\mathbf{b}$ denote the vector formed by stacking all the binary variables b_{jl}, for all $j \in \{1, \ldots, M\}$ and $l \in \{1, \ldots, r\}$, and let $f(\mathbf{b})$ denote the objective function in equation 16. Then, following Hansen (1979), an equivalent optimization problem can be obtained by adding a squared penalty term of the equality constraint to the objective function as follows:

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{b}) + \lambda \left(\sum_{j=1}^{M} \sum_{l=1}^{r} b_{jl} 2^{l-1} - N \right)^2 \\ &\text{subject to} \quad b_{jl} \in \{0, 1\} \ , \ \forall\, j \in \{1, \ldots, M\} \ , \ \forall\, l \in \{1, \ldots, r\} \ , \end{aligned} \tag{C-1}$$

where λ is any suitably chosen constant that satisfies $\lambda \geq \bar{f}$, with $\bar{f}$ being the solution to the following optimization problem

$$\begin{aligned} &\text{maximize} \quad f(\mathbf{b}) \\ &\text{subject to} \quad b_{jl} \in \{0, 1\} \ , \ \forall\, j \in \{1, \ldots, M\} \ , \ \forall\, l \in \{1, \ldots, r\} \ . \end{aligned} \tag{C-2}$$

The optimization problem thus obtained in equation C-1 is now in QUBO form. Notice that although solving for $\bar{f}$ itself involves solving another QUBO in equation C-2, one does not need to do so to obtain a λ satisfying $\lambda \geq \bar{f}$. One such choice is given by observing that, p and L_k having the same sign in this experiment geometry,

$$\begin{aligned} &\sum_{k=1}^{K} \left[\frac{1}{v_{\max}^2} \left(L_k - \sum_{j=1}^{M} \sum_{l=1}^{r} \alpha_{kj} b_{jl} 2^{l-1} \right)^2 + \left(T_k - \sum_{j=1}^{M} \sum_{l=1}^{r} \beta_{kj} b_{jl} 2^{l-1} \right)^2 \right] \\ &\leq \sum_{k=1}^{K} \left[\frac{L_k^2}{v_{\max}^2} + \frac{1}{v_{\max}^2} \left(\sum_{j=1}^{M} \sum_{l=1}^{r} \alpha_{kj} 2^{l-1} \right)^2 + T_k^2 + \left(\sum_{j=1}^{M} \sum_{l=1}^{r} \beta_{kj} 2^{l-1} \right)^2 \right] \\ &\leq \sum_{k=1}^{K} \left[\frac{L_k^2}{v_{\max}^2} + \frac{2^{2r}}{v_{\max}^2} \left(\sum_{j=1}^{M} \alpha_{kj} \right)^2 + T_k^2 + 2^{2r} \left(\sum_{j=1}^{M} \beta_{kj} \right)^2 \right] \ , \end{aligned} \tag{C-3}$$

and therefore,

$$\lambda = \sum_{k=1}^{K} \left[\frac{L_k^2}{v_{\max}^2} + \frac{2^{2r}}{v_{\max}^2} \left(\sum_{j=1}^{M} \alpha_{kj} \right)^2 + T_k^2 + 2^{2r} \left(\sum_{j=1}^{M} \beta_{kj} \right)^2 \right] \tag{C-4}$$

is a valid choice. However, if λ is too large, the objective function in equation C-1 is dominated by the penalty term, and numerical optimization algorithms end up completely ignoring the original objective function that we wanted to minimize. To avoid this issue, it is better to choose a value of λ that is as small as possible that still guarantees that a global minimum of equation C-1 satisfies the constraints. For example, notice that we can

drop any constant term in the objective function $f(\mathbf{b})$ in equation C-1 without changing the optimization problem. This allows us to improve the bound in equation C-4 to

$$\lambda = \sum_{k=1}^{K} \left[\frac{2^{2r}}{v_{\max}^2} \left(\sum_{j=1}^{M} \alpha_{kj} \right)^2 + 2^{2r} \left(\sum_{j=1}^{M} \beta_{kj} \right)^2 \right] \ . \tag{C-5}$$

Other choices for the parameter λ are also possible. We illustrate two such possibilities. The starting point of these choices is the observation that because all the variables b_{jl} are binary, we have the identity $b_{jl} = b_{jl}^2$, and therefore the objective function in the optimization problem of equation 16 can be expressed as a quadratic form if we ignore the constant terms, as follows:

$\mathbf{b}^T Q \mathbf{b}$, where Q is a symmetric matrix of dimensions $Mr \times Mr$ given by

$$Q_{r(j-1)+l,r(j'-1)+l'} = \begin{cases} 2^{l+l'-2} \sum\limits_{k=1}^{K} \left(\frac{\alpha_{kj}\alpha_{kj'}}{v_{\max}^2} + \beta_{kj}\beta_{kj'} \right) \text{ if } j \neq j', \text{ or } l \neq l', \text{else} \\ \sum\limits_{k=1}^{K} \left[\frac{1}{v_{\max}^2} \left(2^{2l-2}\alpha_{kj}^2 - 2^l L_k \alpha_{kj} \right) + \left(2^{2l-2}\beta_{kj}^2 - 2^l T_k \beta_{kj} \right) \right] \end{cases} \tag{C-6}$$

for all $j, j' \in \{1, \ldots, M\}$ and $l, l' \in \{1, \ldots, r\}$.

The first possibility then follows by noticing that as $\mathbf{b}$ is a binary vector, we have

$$\mathbf{b}^T Q \mathbf{b} \leq |\mathbf{b}^T Q \mathbf{b}| \leq \sum_{j,j'=1}^{M} \sum_{l,l'=1}^{r} |Q_{r(j-1)+l,r(j'-1)+l'}| \ . \tag{C-7}$$

The second possibility follows from the inequality $\mathbf{b}^T Q \mathbf{b} \leq \lambda_{\max}(Q) \mathbf{b}^T \mathbf{b} \leq \lambda_{\max}(Q) Mr$, where $\lambda_{\max}(Q)$ is the largest eigenvalue of the matrix Q. Thus, the following choices of λ are also valid choices

$$\lambda = \sum_{j,j'=1}^{M} \sum_{l,l'=1}^{r} |Q_{r(j-1)+l,r(j'-1)+l'}| \text{ , and } \lambda = \lambda_{\max}(Q) Mr \ . \tag{C-8}$$

Equivalent Ising spin-glass Hamiltonian

To solve the optimization problem in equation C-1 using quantum annealing, we introduce the spin variables $s_{jl} \in \{-1, 1\}$ given by

$$s_{jl} = 2b_{jl} - 1 \ , \ \forall\, j \in \{1, \ldots, M\} \ , \ \forall\, l \in \{1, \ldots, r\} \, . \tag{C-9}$$

Let $\mathbf{s}$ denote the vector formed by stacking all the spin variables s_{jl}. The Ising Hamiltonian $H_f(\mathbf{s})$, whose ground state encodes the solution to equation C-1 is obtained by directly substituting the spin variables s_{jl} into the objective function yielding

$$\begin{aligned} H_f(\mathbf{s}) = & \sum_{j,j'=1}^{M} \sum_{l,l'=1}^{r} 2^{l+l'-4} \left[\lambda + \sum_{k=1}^{K} \left(\frac{\alpha_{kj}\alpha_{kj'}}{v_{\max}^2} + \beta_{kj}\beta_{kj'} \right) \right] s_{jl} s_{j'l'} \\ & - \sum_{j=1}^{M} \sum_{l=1}^{r} 2^{l-1} \left[\lambda N' + \sum_{k=1}^{K} \left(\frac{L_k' \alpha_{kj}}{v_{\max}^2} + T_k' \beta_{kj} \right) \right] s_{jl} \ , \end{aligned} \tag{C-10}$$

where we have introduced the quantities L'_k, T'_k and N' defined as follows:

$$\begin{aligned} L'_k &= L_k - \frac{1}{2}(2^r - 1)\sum_{j=1}^{M} \alpha_{kj} \\ T'_k &= T_k - \frac{1}{2}(2^r - 1)\sum_{j=1}^{M} \beta_{kj} \\ N' &= N - \frac{M}{2}(2^r - 1) \ , \end{aligned} \tag{C-11}$$

and we have neglected the constant terms.

APPENDIX D

Full QUBO optimization problem

We present a full QUBO problem formulation for Snell tomography, that is derived from the optimization problem in (16) by additionally discretizing the ray parameters. Let us discretize the ray parameter range $[0, 1/v_{\max})$ into P uniformly spaced values given by the set $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_P\}$, defined as $\hat{p}_q = (q-1)/Pv_{\max}$. The parameter P is used to control the level of discretization that we want, which in turn will affect the accuracy of the solutions. The key step in getting a full QUBO problem formulation is to allow each ray to take all possible values in $\hat{P}$ and for each material. To achieve this goal, we start by defining $\hat{\alpha}_{qj}$ and $\hat{\beta}_{qj}$ as follows

$$\hat{\alpha}_{qj} = \frac{\delta \hat{p}_q v_j}{\sqrt{1 - \hat{p}_q^2 v_j^2}} \ , \ \hat{\beta}_{qj} = \frac{\delta}{v_j\sqrt{1 - \hat{p}_q^2 v_j^2}} \ , \tag{D-1}$$

for all $j = 1, \ldots, M$ and $q = 1, \ldots, P$. We next define indicator variables $\hat{x}_{kq}$ as

$$\hat{x}_{kq} = \begin{cases} 1 & \text{if ray } k \text{ has ray parameter } \hat{p}_q \\ 0 & \text{otherwise.} \end{cases} \tag{D-2}$$

To ensure that each ray gets assigned an unique $\hat{p}_q$ while propagating through each material, we require that the following constraints be satisfied

$$\sum_{q=1}^{P} \hat{x}_{kq} = 1 \ , \ \forall\, k = 1, \ldots, K. \tag{D-3}$$

With constraint (D-3) in place, we can now write the objective function in (16) as follows

$$J = \sum_{k=1}^{K}\sum_{q=1}^{P} \hat{x}_{kq} \left[\frac{1}{v_{\max}^2} \left(L_k - \sum_{j=1}^{M}\sum_{l=1}^{r} \hat{\alpha}_{qj} b_{jl} 2^{l-1} \right)^2 + \left(T_k - \sum_{j=1}^{M}\sum_{l=1}^{r} \hat{\beta}_{qj} b_{jl} 2^{l-1} \right)^2 \right] , \tag{D-4}$$

and notice that it is a quadratic function in the variables $\hat{x}_{kq}$ and b_{jl}, which is clear from the fact that $b_{jl}^2 = b_{jl}$ because the variables b_{jl} are binary. Thus, the full QUBO optimization

problem can now be written as

$$
\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{K}\sum_{q=1}^{P}\hat{x}_{kq}\left[\frac{1}{v_{\max}^2}\left(L_k-\sum_{j=1}^{M}\sum_{l=1}^{r}\hat{\alpha}_{qj}b_{jl}2^{l-1}\right)^2+\left(T_k-\sum_{j=1}^{M}\sum_{l=1}^{r}\hat{\beta}_{qj}b_{jl}2^{l-1}\right)^2\right] \\
\text{subject to} \quad & b_{jl}\in\{0,1\}\ ,\ \forall\, j\in\{1,\ldots,M\}\ ,\ \forall\, l\in\{1,\ldots,r\}\ , \\
& \hat{x}_{kq}\in\{0,1\}\ ,\ \forall\, k\in\{1,\ldots,K\}\ ,\ \forall\, q\in\{1,\ldots,P\}\ , \\
& \sum_{j=1}^{M}\sum_{l=1}^{r}b_{jl}2^{l-1}=N\ , \\
& \sum_{q=1}^{P}\hat{x}_{kq}=1\ \ ,\ \ \forall\, k=1,\ldots,K.
\end{aligned}
\tag{D-5}
$$

As before the equality constraints can be removed by adding them as penalty terms, for example as outlined in Appendix C. The number of binary variables involved in solving optimization problem in (D-5) is $KP+Mr$.