BATTLESHIP, TOMOGRAPHY, AND QUANTUM ANNEALING

W. RILEY CASPER AND TAYLOR GRIMES

Abstract. The classic game of Battleship involves two players taking turns attempting to guess the positions of a fleet of vertically or horizontally positioned enemy ships hidden on a 10×10 grid. One variant of this game, commonly referred to as Battleship Solitaire, Bimaru, or Yubotu, considers the game with the inclusion of *X-ray data*, represented by knowledge of how many spots are occupied in each row and column in the enemy board.

This paper considers the *Battleship puzzle problem*: the problem of reconstructing an enemy fleet from its X-ray data. We generate non-unique solutions to Battleship puzzles via certain reflection transformations akin to Ryser interchanges. Furthermore, we demonstrate that solutions of Battleship puzzles may be reliably obtained by searching for solutions of the associated classical binary discrete tomography problem which minimize the discrete Laplacian. We reformulate this optimization problem as a QUBO problem and approximate solutions via a simulated annealer, emphasizing the future practical applicability of quantum annealers to solving discrete tomography problems with predefined structure.

1. Introduction

In Battleship, two opposing players secretly position fleets of ships of predetermined lengths horizontally or vertically on a 10×10 grid. Then each player takes turns guessing the locations of enemy ships. This paper considers a natural extension of the game wherein each player possesses X-ray telemetry which gives them knowledge of the row or column sums of occupied ship positions in their opponent's board, as in Figure 1. The problem then becomes how to leverage the X-ray data in order to discern the locations of the enemy fleet.

The Battleship puzzle problem of determining a full Battleship board from its row and column sums, possibly with some additional hints about ship positions, is commonly referred to as Battleship Solitaire. The origin of Battleship Solitaire is attributed to Jaime Poniachik in 1982. In practice, these puzzles often feature additional information such as locations of pieces of ships, which force solutions of the battleship puzzle to be unique. The problem of determining whether a battleship puzzle has a solution, unique or otherwise, is known to be NP-complete [5]. Here, we focus on the specific case when we have knowledge of the row and column sums *only*, so that in general solutions may not be unique.

1.1. Binary Tomography and QUBO.

Inspired by Gritzmann [10], our formulation of the Battleship puzzle problem in terms of Battleship with X-rays underscores the puzzle's relation with tomography, specifically binary tomography. In classical tomography, X-rays are sent through a solid object at various angles. By measuring the intensity of the X-ray exiting the object relative to its original intensity, we can quantify the average density of the object along the X-ray's trajectory. Tomography then deals with the

Key words and phrases. binary discrete tomography, quantum computing, D-Wave, Battleship, QUBO.

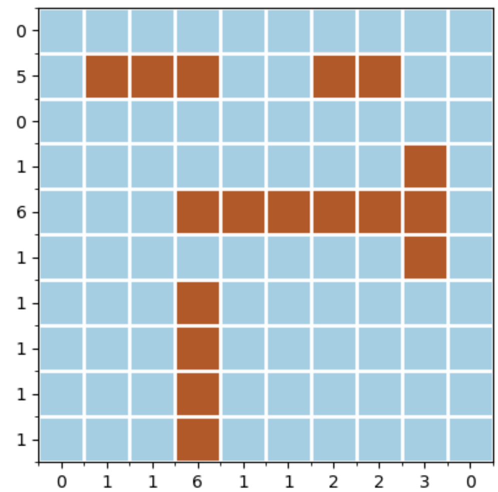

FIGURE 1. Row and column sums for a Battleship fleet. In Battleship with X-rays, X-ray telemetry reveals the density of the opponents fleet in horizontal and vertical directions, represented by the sum of the occupied spaces in each row and column in the board.

mathematical process of reconstructing an image of the internal structure of the object from this density data.

In binary tomography, we consider a discrete version of the above tomography problem. Our fictitious X-rays probe information about a $m \times n$ binary matrix A, telling us the row and column sums. The original problem in binary tomography is then to construct an $m \times n$ binary matrix with specific row and column sums. Ryser, and independently Gale, determined necessary and sufficient conditions for the existence of a binary matrix with specified row or column sums, along with a polynomial time algorithm for constructing a solution [2, 3, 4]. Furthermore, different solutions are related by a series of transformations we refer to as Ryser interchanges. See [1] for a comprehensive treatment.

We can interpret a Battleship puzzle as a binary tomography problem, wherein the locations of ships are represented by ones in a 10×10 binary matrix. However, our problem differs from the one solved by Ryser in that we must construct solutions corresponding to placements of a Battleship fleet. Thus it must necessarily feature ships of predetermined lengths, ie. certain contiguous lines of 1's. This presents two problems:

(1) there are many solutions of the binary tomography problem which do not represent fleet positions;
(2) Ryser interchanges break up ships, leading to boards which do not represent fleet positions.

The second problem is addressed in Section 2, where we create a generalization of Ryser interchanges which does allow us to preserve the property of representing a Battleship fleet. In particular this shows that different Battleship fleets can have the same row and column sums, so that the Battleship puzzle problem does not have a unique solution unless other constraints are imposed.

The first problem is more serious. In practice, it makes Ryser's methods of constructing a solution infeasable. While Ryser gives us a method of constructing a canonical binary matrix with the desired row and column sums, this canonical solution is unlikely to correspond to any possible Battleship fleet position. Furthermore, in a typical case there may be an enormous amount of binary matrices with the given row and column sums. Calculating all such solutions and searching among them for those matrices which could be fleet positions can require large amounts of memory and multiple days of computation time, and thus is computationally impractical.

To fix this issue, we explore the set $\mathfrak{U}(R, S)$ of all $m \times n$ binary matrices with row sums $R = (r_1, \ldots, r_m)$ and column sums $S = (s_1, \ldots, s_n)$. The size of the set $\mathfrak{U}(R, S)$ can be very large and determining estimates for its size is an interesting combinatorial question [11, 12]. Via numerical exploration in Section 3, we find that the binary matrices corresponding to Battleship fleets tend lie close to the minimum of the sum of the squares of the discrete Laplacian. Intuitively, this is because when we force the solution to consist of specific contiguous straight lines we impose a lot of structure and minimize the presence of edges in the binary matrix. Since the discrete Laplacian works as a sharpening mask in image processing, this corresponds to having minimal Laplacian values. Thus our strategy is to find those elements in $\mathfrak{U}(R, S)$ which are near the minimal Laplacian values.

The strategy outlined in the previous paragraph can be reformulated as a binary optimization problem. In Section 4, we translate it into a quadratic unconstrained binary optimization (QUBO) problem, ie. the problem of finding a binary vector $\vec{x}$ which minimizes $\vec{x}^T Q \vec{x}$ for some fixed matrix Q. We may choose Q such that values $\vec{x}$ minimizing $\vec{x}^T Q \vec{x}$ correpond precisely to solutions of the discrete tomography problem which minimize the value of the sum of the squares of the discrete Laplacian. In Section 5, we approximate solutions of this QUBO problem both with standard methods and using a simulated annealer. We demonstrate that the vast majority of randomly generated Battleship puzzle problems may be rapidly solved via both methods.

1.2. **Quantum annealing and Ising models.** A quantum annealer is a type of quantum computer, a computer which uses quantum phenomena such as entanglement and tunneling to perform computations. Specifically, a collection of qubits are arranged in a particular lattice. Each qubit has two possible observable quantum states, spin up $|+1\rangle$ and spin down $|-1\rangle$, and at any particular timie is represented by a wave function $\psi = a|+1\rangle + b|-1\rangle$ where $a, b \in \mathbb{C}$ are complex numbers with $|a|^2$ and $|b|^2$ representing the probabilities of being spin up and spin down, respectively (so that $|a|^2 + |b|^2 = 1$). In the presence of a magnetic field, the qubits will tend to align in various directions, in accordance with a potential energy defined by

$$E(\vec{x}) = -\vec{x}^T J \vec{x} - \sum_{j=1}^{n} \beta_j x_j$$

for some $n \times n$ matrix J and some constants $\beta_1, \ldots, \beta_n$, where n is the number of qubits. Here $\vec{x}$ is a vector whose entries are ± 1, corresponding to the direction of the spin of each qubit. This is referred to as an **Ising model**. By carefully choosing the magnetic field experienced by the qubits inside a quantum annealer, the energy in a quantum annealer can be made to resemble any desired values of J and $\beta_1, \ldots, \beta_n$, though with some constraints based on the network topology of the qubits, the number of qubits, and the presence of noise in the system.

The Ising problem is the problem of finding $\vec{x} \in \{\pm 1\}^n$ minimizing the energy $E(\vec{x})$ in the Ising model. Note that one may easily convert between Ising problems and QUBO problems so that both problems are equivalent. In general such problems are NP-complete and difficult to solve explicitly, though many standard methods of approximating solutions exist. A quantum annealer solves an Ising (or QUBO) problem using physics rather than mathematics. Starting with each qubit in a supercooled state and in the presence of a uniform magnetic field, the whole system will be arranged in a global minimum energy state. Then by gradually evolving the background magnetic field, we can move from a simple equation for the energy to the energy of the quantum system corresponding to a particular Ising problem. When this process is performed slowly enough, each qubit will have a high probability of remaining at or near the minimum energy. Thus by observing the quantum system we obtain good approximations to the solution of the Ising problem. By leveraging quantum phenomena, these approximate solutions are anticipated to be obtained far faster than by classical approaches, especially for large problem sizes.

Presently, quantum annealers are very constrained in terms of the number of qubits. The largest, at the time of this writing, is manufactured by D-Wave Systems and features around 5000 qubits – a slim number when compared to the gigabytes avalable from RAM cards on classical computing systems available to the average consumer. However, in the future we anticipate the existence of quantum annealers with enormous amounts of qubits. Thus encoding complicated mathematical problems like the Battleship puzzle problem is a key step in the path of leveraging this future computing resource. We used the simulated annealer dwave-neal [15] to estimate the performance of our algorithm on an actual quantum annealer. For a comparison of simulated annealing with quantum annealing on the D-Wave, see [14].

2. Stealthy Fleet Positions

In a competitive game of Battleship with X-rays, a player would naturally wish to choose positions for their ships where the X-ray data, the row and column sums, won't give the game away. In other words, we would seek **stealthy fleet positions**, by which we mean Battleship fleets where the row and column sums are shared between many different fleets. However, it is not obvious at all whether stealthy fleet positions exist or the row or column sums determine the fleet positions uniquely.

Perhaps surprisingly, examples of stealthy fleet positions are in great supply. For examples, see Figure 2 which describes a relationship between two different fleet positions with the same X-ray data, and Figure 4 which gives examples of nine different fleets with the same row and column sums. Thus as in the case of binary tomography, the row and column sums are insufficient to specify the precise position of the fleet. The source of this lack of uniqueness in the tomography situation is the existence of Ryser interchanges.

Definition 2.1. Let A be a $m \times n$ binary matrix and suppose that $1 \leq a, c \leq m$ and $1 \leq b, d \leq n$ satisfy $A(a,b) = 1$, $A(c,d) = 1$, $A(c,b) = 0$ and $A(a,d) = 0$. The (a,b,c,d) **Ryser interchange** of A is an $m \times n$ binary matrix B whose values are given by

$$B(j,k) = \begin{cases} 1 - A(j,k), & (j,k) = (a,b), (a,d), (c,b), \text{ or } (c,d) \\ A(j,k), & \text{otherwise.} \end{cases}$$

As one can readily see, a Ryser interchange of a matrix A preserves the values of the row and column sums. Furthermore, Ryser proved any binary matrix whose row and column sums are the same as those of A may be obtained via a sequence of Ryser interchanges.

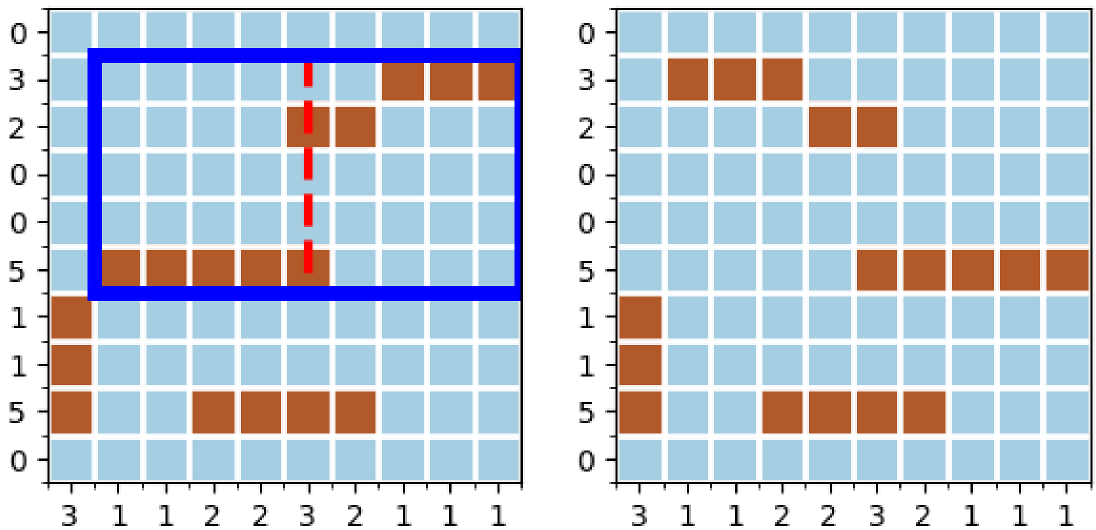

FIGURE 2. Two different Battleship fleets with the same row and column sums. The fleet on the right is obtained by the one on the left by reflecting the indicated submatrix horizontally across the dashed vertical line.

2.1. **Generalizing Ryser interchanges.** In our situation, the problem is complicated by the fact that Ryser interchanges can break up ships, returning binary matrices whose entries cannot possibly represent the positions of a Battleship fleet. To combat this issue, we introduce a generalization of Ryser interchanges which preserve both ships and the row and column sum data. To start, let J_m represent the $m \times m$ anti-diagonal identity and recall that $J_m A$ flips the rows of A up-to-down. Likewise AJ_n flips the columns of A left-to-right.

Definition 2.2. Let A be a $m \times n$ binary matrix and let $\widetilde{A}$ be a $p \times q$ submatrix of A. The submatrix $\widetilde{A}$ is said to be **column sum-symmetric** if the column sums of $\widetilde{A}$ and $J_p\widetilde{A}J_q$ are the same. Likewise, $\widetilde{A}$ is said to be **row sum-symmetric** if the row sums of $\widetilde{A}$ and $J_p\widetilde{A}J_q$ are the same. The **horizontal subreflection of $\widetilde{A}$ in** A is the $m \times n$ matrix B which is the same as A, but with the submatrix $\widetilde{A}$ flipped left-to-right. Likewise the **vertical subreflection of $\widetilde{A}$ in** A is the matrix C which is the same as A, but with the submatrix $\widetilde{A}$ flipped up-to-down.

With these definitions in mind, it is straightforward to prove the following proposition.

Proposition 2.3. *Let A be a $m \times n$ binary matrix and let $\widetilde{A}$ be a $p \times q$ submatrix of A. If $\widetilde{A}$ is column sum-symmetric, then the horizontal subreflection of $\widetilde{A}$ in A has the same row and column sums as A. Likewise, if $\widetilde{A}$ is row sum-symmetric, then the vertical subreflection of $\widetilde{A}$ in A has the same row and column sums as A.*

Now let's restrict ourselves to the situation of a 10×10 binary matrix A. A Battleship fleet is composed of 5 ships: the destroyer, submarine, cruiser, battleship, and carrier whose lengths are $2, 3, 3, 4$, and 5, respectively.

Definition 2.4. Let A be a 10×10 binary matrix. A **fleet realization** for A is a choice of the positions of the five ships in the fleet such that the nonzero entries of A coincide with the positions occupied by the ships.

It is worth noting that in order for A to have a fleet realization, it is necessary but not sufficient for it to have precisely 17 nonzero entries. Therefore the sum of the row sums and the sum of the column sums must both be 17. Moreover, it is possible for A to have multiple fleet realizations, for example, when the destroyer and cruiser are found end-to-end, we will have a decision to make about which sequence of adjacent ones represents the carrier and which represents the destroyer-cruiser combination.

Definition 2.5. Let A be a 10×10 binary matrix with at least one fleet realization, and let $\widetilde{A}$ be a $p\times q$ submatrix. We say $\widetilde{A}$ is **non-bisecting** if there exists at least one fleet realization satisfying the property that any ship whose position overlaps with $\widetilde{A}$ must lie entirely within $\widetilde{A}$.

Thus to find a new fleet position whose row and column sums are the same as the old one, we can search for a non-bisecting column-sum symmetric or row-sum symmetric submatrix $\widetilde{A}$ of A and perform either a horizontal or vertical subreflection. For example the different fleet positions giving rise to the same row and column sums in Figure 2 are connected by a horizontal subreflection.

3. Laplace-minimimizing binary matrices

In this section we describe the discrete Laplacian operator and observe numerically that when we arrange binary matrices with discrete row and column sums by the magnitude of the square of their discrete Laplacians, we tend to find legitimate Battleship fleet positions among the minimal values. Thus to find a Battleship fleet position for a certain set of row or column sums we are motivated to search for those matrices which minimize or nearly minimize the square of the Laplacian.

3.1. Discrete Laplacians.

Consider an $m\times n$ rectangular sublattice $D=\{1,\dots,m\}\times\{1,\dots,n\}\subseteq\mathbb{Z}\times\mathbb{Z}$. The **discrete Laplacian** of a function $f:D\to\mathbb{R}$ is the function $\Delta f:D\to\mathbb{R}$ defined by

$$-\Delta f(j,k)=4f(j,k)-f(j-1,k)-f(j,k-1)-f(j+1,k)-f(j,k+1),$$

where here we interpret $f(j,k)=0$ for $(j,k)\in\mathbb{Z}^2\backslash D$, imposing a Dirichlet-type condition on the value of f outside the domain D. Since an $m\times n$ matrix A may be interpreted readily as function via $f(j,k)=A_{jk}$, we are naturally able to define the Laplacian of a matrix A, which as an abuse of notation we denote by ΔA.

The discrete Laplacian above arises commonly in finite-difference and finite-element numerical methods for partial differential equations. It also arises naturally in image processing, specifically in the context of edge-detection where it plays the role of a simple digital sharpening filter. The sum of the squares of the Laplacian is the same as the ℓ^2-norm squared of the associated vector.

$$\|\Delta f\|^2=\sum_{j,k}\left[4f(j,k)-f(j-1,k)-f(j,k-1)-f(j+1,k)-f(j,k+1)\right]^2.$$

It's worth noting that in the continuous case multivariate functions which minimize the squares of their Laplacians and satisfy some prescribed boundary behavior were studied in [6], where they were shown to be biharmonic functions (ie. $\Delta^2 f=0$) away from interior boundaries of their domains. We could imagine a similar characterization in the discrete situation, except for the complications that (1) we are interested in *binary* functions, and (2) our constraint is a condition on the sum of horizontal and vertical projections and is nonlocal in comparison to prescribing certain boundary values as done in [6]. Note that for us, the boundary values are Dirichlet in the sence that $f(j,k)$ is zero outside the prescribed domain D.

3.2. **Discrete Laplacians of Battleship Fleets.** In this subsection we consider the values of the sums of squares of Laplacians for functions with prescribed row and column sums. As we demonstrate, those solutions minimizing the value of this sum tend to be more ordered, and more often have Battleship fleet realizations. To begin, consider the following example.

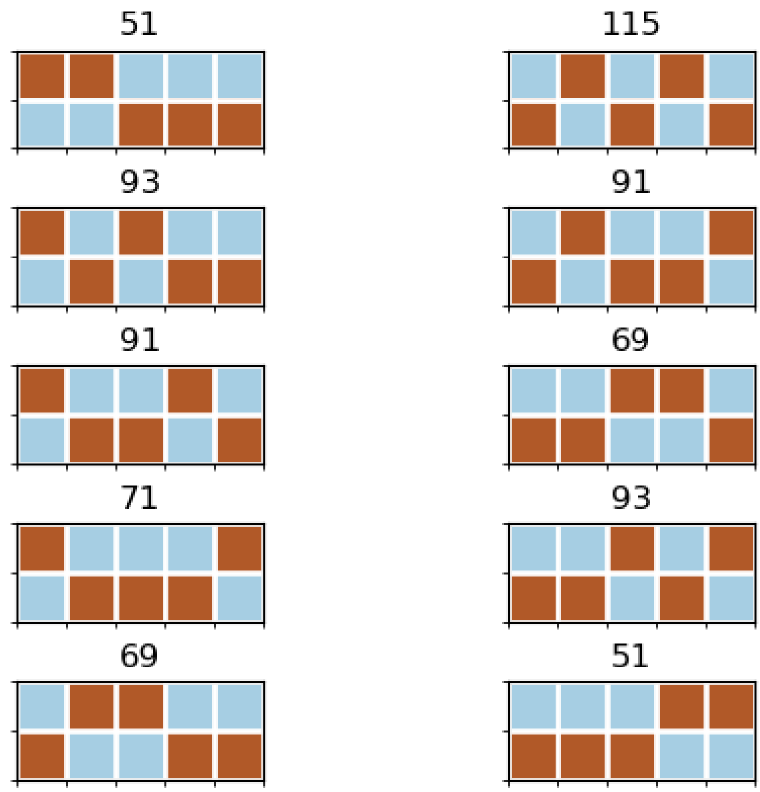

FIGURE 3. The 2×5 binary matrices whose column sums are all 1 and whose row sums are 2 and 3, respectively. The sum of the squares of the Laplacian of the matrix is indicated above each arrangement. The potential Battleship fleet positions correspond with the smallest sum values.

Example 3.1. Consider the placement of a 2×1 destroyer and a 3×1 submarine inside a 2×5 grid such that the column sums are all 1 and the row sums are 2 and 3, respectively. There are ten such binary matrices with these specific row and column sums and their values along with the sums of the squares of their Laplacians can be found in Figure 3 below. The arrangements with the smallest Laplacian squared sums correspond to the actual possible fleet positions.

What we observe in the simpler 2×5 case also carries over to the full board. In fact, what we find by means of direct numercal experimentation is that generically the binary matrices with fleet realizations typically have minimal or near-minimal values for the sums of the squares of their Laplacians. Intuitively this makes sense because

- straight lines of ones in the binary matrix tend to have smaller Laplacians than shapes with many turns
- more connected fleet positions tend to have smaller Laplacians than fleet positions with many connected components, particularly those with stranded single entries

As a particular case, we explore the set of Battleship fleets featured in Figure 4. Each

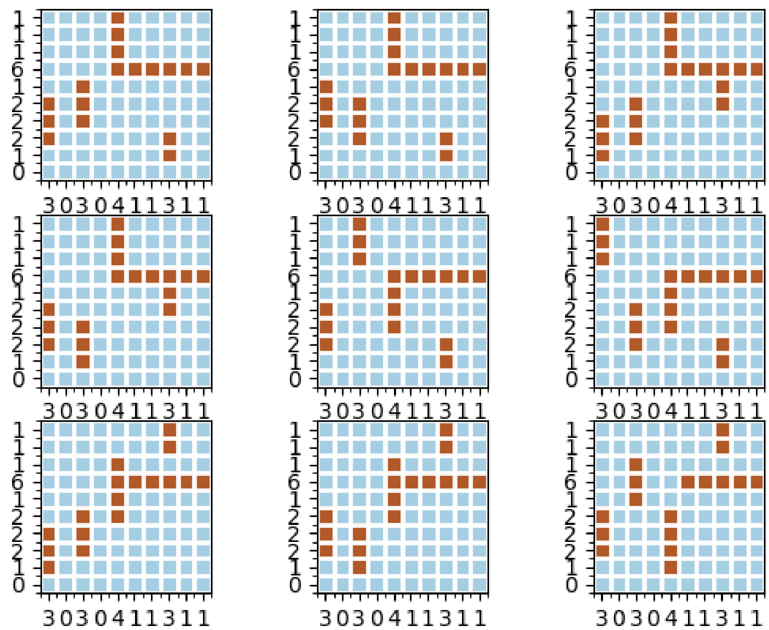

FIGURE 4. Fleets for a particular row and column sum which are four Ryser interchanges from the starting fleet. The starting fleet is featured in the upper left corner.

fleet has the same row and column sums. We use a computer to determine the set $\mathfrak{U}(R, C)$ of all binary matrices with these row and column sums. The ℓ^2 norm squared of the entries of $\mathfrak{U}(R, C)$ form a Gaussian distribution with mean value $\mu = 248.154$ and standard deviation $\sigma = 21.935$, as indicated in Figure 5. The squared norm values of the binary matrices corresponding to the fleet positions are indicated by the vertical dashed lines. Notably all lie at the far left part of the distribution, at or near the minimum value and more than two standard deviations from the mean.

4. Reconstructing Fleet Positions

In this section, we describe our method of reconstructing fleet positions from knowledge of the row and column sums. Our strategy is to reformulate the problem as a QUBO problem of finding the 10×10 binary matrices with the correct row or column sums whose values are near the minimum of the square of the two-dimensional discrete laplacian.

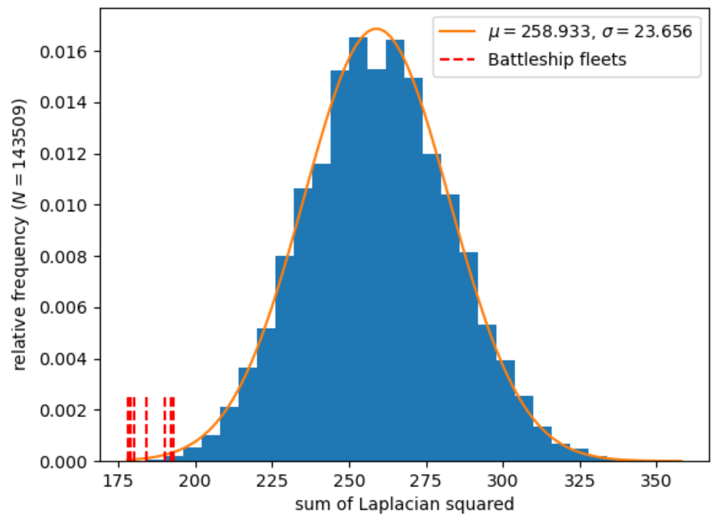

FIGURE 5. Histogram of the sums of squares of Laplacians for all binary matrices which are four Ryser interchanges away from the starting fleet along with a curve fit to a normalized Gaussian distribution. The binary matrices with fleet realizations are indicated by the dashed vertical lines and lie to the extreme left of the distribution around 3σ from μ.

4.1. **Converting to QUBO.** Let Q be an $n \times n$ matrix, L an $m \times n$ matrix and $\vec{b}$ length m vector satisfying $L\vec{x} = \vec{b}$ for some $\vec{x} \in \{0,1\}^n$. Consider the linearly constrained binary optimization problem of minimizing $\vec{x}^T Q \vec{x}$ subject to the constraint that $L\vec{x} = \vec{b}$. It turns out that such a constrained problem can be converted into a QUBO problem in a standard way.

Theorem 4.1. *Let Q and L be as above and define*

$$Q_{lin} = L^T L - 2\,diag(L^T \vec{b}).$$

For ϵ small enough, solutions of the QUBO problem

$$\text{minimize } \vec{x}^T (Q_{lin} + \epsilon Q)\vec{x}, \quad \vec{x} \in \{0,1\}^n \tag{1}$$

will also be solutions of the optimization problem with a linear constrant

$$\text{minimize } \vec{x}^T Q \vec{x}, \quad \vec{x} \in \{0,1\}^n,\ L\vec{x} = \vec{b} \tag{2}$$

Proof. First note that

$$\vec{x}^T L^T L \vec{x} - 2\vec{x}^T L^T \vec{b} + \vec{b}^T \vec{b} = (L\vec{x} - b)^T (L\vec{x} - \vec{b}) \geq 0$$

with equality if and only if $L\vec{x} = \vec{b}$. Since $\vec{x}$ is binary, $\vec{x}^T \vec{L}^T \vec{b} = \vec{x}^T \text{diag}(\vec{L}^T \vec{b})\vec{x}$. Thus the binary solutions of $L\vec{x} = \vec{b}$ coincide with solutions of the optimization problem

$$\text{minimize } \vec{x}^T Q_{\text{lin}} \vec{x}, \quad \vec{x} \in \{0,1\}^n.$$

Thus if r is the spectral gap of Q_{lin} and R is the spectral radius of Q, then for $L\vec{x} \neq \vec{b}$ and $\epsilon = r/10R$ we see that the minimum solution of

$$\text{minimize } \vec{x}^T (Q_{\text{lin}} + \epsilon Q)\vec{x}, \quad \vec{x} \in \{0,1\}^n$$

will necessarily satisfy $\vec{x}^T Q_{\text{lin}} \vec{x} = 0$ and thus will be a solution of the linearly constrained problem above. $\square$

Thus to solve the problem of finding binary matrices with certain row and column sums which minimize the Laplacian, we can convert the associated linear system to a QUBO problem with matrix Q_{lin}, construct a matrix Q_{lap} encoding the Laplacian and solve the QUBO problem for the matrix

$$Q = Q_{\text{lin}} + \epsilon Q_{\text{lap}}$$

for sufficienty small ϵ. We discuss this in detail in the next subsection.

4.2. **Laplacian-minimizing solutions.** To calculate the Laplacian-minimizing 10×10 binary matrices A with prescribed row and column sums, we start by expressing A as a binary vector $\vec{x}$ of length 100 whose entries are $x_{10(j-1)+k} = A_{jk}$ with $1 \leq j, k \leq 10$. Taking the row and column sums of A is then equivalent to multiplying $\vec{x}$ by a 20×100 matrix L whose entries are

$$L_{m,10(j-1)+k} = \begin{cases} 1, & 1 \leq m \leq 10 \text{ and } n = k \\ 1, & 11 \leq m \leq 20 \text{ and } j = m - 10 \\ 0, & \text{otherwise} \end{cases}, \quad 1 \leq j, k \leq 10$$

In particular, if $\vec{r}$ and $\vec{c}$ are column vectors whose entries are the row and column sums of A (in order from top to bottom or left to right) then $L\vec{x} = \binom{\vec{r}}{\vec{c}}$.

The discrete Laplacian of a matrix A is a new matrix ΔA whose size is the same as A and whose entries are

$$(\Delta A)_{j,k} = 4A_{j,k} - A_{j-1,k} - A_{j+1,k} - A_{j,k-1} - A_{j,k+1}$$

where entries $A_{m,n}$ outside the bounds of $1, \ldots, 10$ are taken to be zero. This corresponds to the multiplication of $\vec{x}$ by a 100×100 matrix, which as an abuse of notation we also denote by Δ:

$$\Delta_{10(s-1)+t,10(j-1)+k} = \begin{cases} 4, & s = j \text{ and } t = k \\ -1, & s = j \text{ and } |t - k| = 1 \\ -1, & t = k \text{ and } |s - j| = 1 \\ 0, & \text{otherwise} \end{cases}, \quad 1 \leq j, k \leq 10.$$

In this form, the sum of the squares of the Laplacian of a matrix A is equal to the product $\vec{x}^T(\Delta^T \Delta)\vec{x}$ where $\vec{x}$ is the vector version of A.

Thus the search for binary matrices A with presecribed row sums and column sums given by $R = [r_1 \ \ldots \ r_{10}]$ and $C = [c_1 \ \ldots \ c_{10}]$, respectively, which minimize the square-sum of the discrete Laplacian is equivalent to the QUBO problem

$$\text{minimize} \quad \vec{x}^T(Q_{\text{lin}} + \epsilon Q_{\text{lap}})\vec{x}, \quad x \in \{0,1\}^{100} \tag{3}$$

where here $Q_{\text{lap}} = \Delta^T \Delta$ and

$$Q_{\text{lin}} = L^T L - 2\text{diag}([r_1 \ \ldots \ r_{10} \ c_1 \ldots \ c_{10}]L).$$

In practice the value of ϵ is a parameter that we can toggle to potentially improve performance. The results of our numerical experiements below were obtained using a value of $\epsilon = 0.005$.

4.3. **Numerical Experiments.** In this subsection we demonstrate the efficacy of our algorithm for recovering Battleship fleet positions with prescribed row and column sums.

We explore the ability of our QUBO reformulation to solve Battleship puzzles by leveraging two different QUBO solvers: a classical tabu solver and a simulated annealer. Starting with a randomly generated Battleship fleet, we construct the QUBO problem associated to the given row and column sums, as described in the previous section. Then we try to reconstruct a fleet with the same row and column sums using one of these QUBO solvers. Note that we are not interested in whether or not we recover the specific Battleship fleet we started with, since in practice our generalized Ryser interchanges allow us to quickly move from a recovered fleet position to the original one. As shown in Figure 6 below, both algorithms are able to successfully reconstruct a Battleship fleet position with the desired row and column sums for over 90% of the randomly generated fleets.

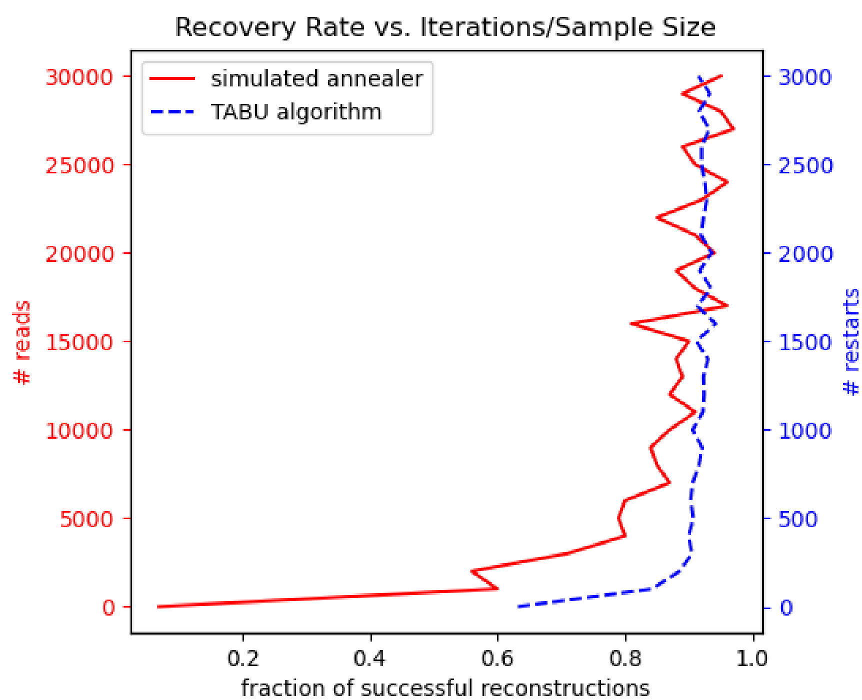

Figure 6. Performance of tabu search versus number of iterations on reconstructing randomly generated fleets from their row and column sums. For ≥ 200 iterations, upwards of 92% of randomly generated fleets are able to be reconstructed via this search method.

4.4. **Numerical methods.** To investigate the practicality of our method of reconstructing fleet positions via solving QUBO problems, we used two common metaheuristic solving methods: tabu and simulated annealing. The tabu search method approximates a solution of the QUBO problem of minimizing the quadratic potential energy $E(\vec{x}) = \vec{x}^T Q\vec{x}$ by iteratively updating an approximate solution and then moving to the adjacent vector with lowest potential energy, excluding a certain list of tabu or forbidden states. After many iterations, the presence of a large set of forbidden states may cause iterations to *increase* rather than decrease the potential energy, allowing the algorithm to escape local minima in the potential energy and more successfully obtain global minima. For details and applications, see [8, 9].

We used simulated annealing as a substitute for an actual quantum annealer, in order to get a rough estimate of how solvable our system would be on a quantum computer. A simulated annealer works by starting with an initial binary vector $\vec{x}$ and then jumping to a random neighboring vector which is either of lower energy or potentially of higher energy but with a probability which decreases based on the iteration. For an introductory account, see [16].

For our work, we use the tabu search algorithm implemented by the QBSolv library in Python [7], which is based on a multistart tabu search method described in [13] using randomly generated initial states. The main parameter we adjust in this algorithm is the number of times we restart the search algorithm using a different randomly generated initial state. The algorithm then returns a list of the minimum energy states found after each restart, which is typically much smaller than the number of restarts do to the presence of duplicates. We also used the simulated annealing python library dwave-neal available from D-Wave Systems [15], where the main adjustment parameter is the number of reads, with each read representing a separate run of the simulated annealing algorithm from a different starting position. We consider a reconstruction to be successful if one of the fleets returned has a Battleship fleet realization. As shown in Figure 6, we are able to obtain a Battleship fleet with the desired row and column sums over 92% of the time for randomly generated Battleship fleets, as long as we use 1000 algorithm restarts in the tabu search algorithm. The behavior of the sumulated annealer is similarly successful and compares well with the tabu algorithm as long as we use 10000 algorithm reads.

5. Summary

In this paper, we considered the Battleship puzzle problem from the point of view of discrete tomography and quantum computing. We demonstrated the existence of different battleship fleets with the same discrete tomographic data and showed how to create new fleets with identical tomographic data from old ones via generalized Ryser interchanges. We demonstrated empirically that Battleship fleets tend to have lower values for the norm-squared of their Laplacians, compared to other discrete binary matrices with the same row and column sums. Using this, we constructed a QUBO-based algorithm for reconstructing Battleship fleets from their row and column sum data. Lastly, we demonstrated the success of this algorithm using both tabu-based and simulated annealing search algorithms. The latter provides evidence that similar sort of Laplacian-minimizing QUBO problems could be successfully solved on a quantum annealer.

References

[1] Gabor T. Herman and Attila Kuba *Discrete tomography: Foundations, algorithms, and applications* Business Media, 2012

[2] H. J. Ryser *Matrices of zeros and ones*, Bull. Amer. Math. Soci. 66(6): 442–464 (1960).

[3] H. J. Ryser *Combinatorial properties of matrices of zeros and ones* Canad. J. Math 9, 371–377 (1957).

[4] David Gale *A theorem on flows in networks*, Pacific J. Math. 7, 1073–1082 (1957).

[5] Merlijn Sevenster *Battleships as Decision Problem*, ICGA Journal. Vol. 27, No. 3, pp. 142–149 (2004).

[6] Kounchev, Ognyan *Minimizing the Laplacian of a function squared with prescribed values on interior boundaries-Theory of polysplines*, Trans. AMS 350.5 (1998): 2105-2128.

[7] Michael Booth, Steven P. Reinhardt, and Aidan Roy *Partitioning optimization problems for hybrid classical/quantum execution*, `https://github.com/dwavesystems/qbsolv/blob/master/qbsolv_techReport.pdf`, August 2017.

[8] Fred Glover *Tabu Search Part 1*, ORSA Journal on Copmuting. 1 (2): 190-206 (1989).
[9] Fred Glover *Tabu Search Part 2*, ORSA Journal on Copmuting. 2 (1): 4-32 (1990).
[10] Peter Gritzmann *Discrete tomography: From battleship to nanotechnology*, Mathematics Everywhere 81-98 (2010).
[11] Alexander Barvinok *On the number of matrices and a random matrix with prescribed row and column sums and 0-1 entries*, Advances in Mathematics 224.1 316-339 (2010).
[12] Ben Johnsen and Eldar Straume *Counting binary matrices with given row and column sums*, Mathematics of computation 48.178 737-750 (1987).
[13] Gintaras Palubeckis *Multistart tabu search strategies for the unconstrained binary quadratic optimization problem*, Annals of Operations Research 131.1 259-282 (2004).
[14] Koshka, Yarosla and Mark A. Novotny *Comparison of D-wave quantum annealing and classical simulated annealing for local minima determination*, IEEE Journal on Selected Areas in Information Theory 1.2 515-525 (2020).
[15] D-wave Systems *dwave-neal*, `https://github.com/dwavesystems/dwave-neal`, July 21, 2021.
[16] Aarts, E. H. L. and P. J. M. Van Laarhoven Simulated annealing: an introduction., Statistica Neerlandica 43.1: 31-52 (1989).

Department of Mathematics, California State University, Fullerton, CA 92831, U.S.A.
Email address: `wcasper@fullerton.edu`

Department of Mathematics, California State University, Fullerton, CA 92831, U.S.A.
Email address: `taylorgrimes2002@csu.fullerton.edu`