Optimizing Quantum Annealing to Advance Graph Coloring Algorithms

Anooshka Pendyal

Abstract

Quantum computing is a rapidly growing field that applies quantum mechanics to computer science and its algorithms, and it has many applications, including optimization problems, where the goal is to obtain an optimal solution to a multivariable problem. Quadratic unconstrained binary optimization problems can represent combinatorial optimization problems, which have a wide range of applications from artificial intelligence to finance. Thus, Sudoku problems, which can be classified as graph-coloring problems, can be represented as optimization problems so that they can be solved with quantum annealing, the process of using quantum fluctuations to solve optimization problems. This project aimed to use Sudoku as a method of evaluating the performance of the latest quantum annealer and to develop a novel method to reduce the number of variables in a QUBO formulation representing Sudoku problems. Creating a more efficient formulation has applications in artificial intelligence and data mining. First, the Sudoku problem was formulated as an objective function so that it could be converted into a QUBO formulation and the variables could be reduced. Then, test cases of various difficulties were created and each was solved with both quantum annealing and simulated annealing. The experiment showed that both quantum annealing and simulated annealing can be used to effectively solve Sudoku problems, although they do have limitations when the number of missing digits from the puzzle exceeds a certain amount. The variable reduction technique was also proven to be effective as it reduced 85 to 99% of variables in all test cases.

Introduction

Quantum computing is an emerging but rapidly developing sector of technology that harnesses the powers of quantum mechanics. The fundamental unit of a quantum computer is a qubit, which can represent the values of 0, 1, or some combination between, in a concept known as superposition, as opposed to a classical bit, which can represent only 0 or 1 (National Academies of Sciences et al., 2019). This allows the quantum computer to represent an exponentially larger amount of data with a small number of qubits (National Academies of Sciences et al., 2019). There are many ways of physically representing qubits, such as single atoms and single electrons or whole systems like in complex superconducting electrical circuits (Preskill, 2018). Logical qubits, in contrast, are emulated more robust and stable qubits (National Academies of Sciences et al., 2019). Quantum optimization employs the advantages of quantum computing while also utilizing the powerful global search ability of intelligent optimization algorithms (Li et al., 2020). Optimization problems are often formulated as minimization problems, where the goal is to obtain the smallest possible error and the optimal solution has the smallest error.

Quadratic unconstrained binary optimization (QUBO) problems are mathematical formulations that can be used to represent a wide variety of problems in the field of combinatorial optimization (Glover et al., 2019). Combinatorial optimization problems focus on finding an optimal solution based on a large number of decisions that yield a corresponding objective function value, such as a cost (Glover et al., 2019), and have a wide range of applications from artificial intelligence to finance (Kochenberger et al., 2014). QUBO problems are NP-hard (Kochenberger et al., 2014) and, due to their relatedness to Ising problems, can be used to represent problems solved through quantum annealing (Glover et al., 2019). Quantum annealing is the process of using quantum fluctuations to solve optimization problems. The properties of quantum mechanics can be harnessed to solve multivariable optimization problems framed as energy minimization problems (Morita and Nishimori, 2008). Similarly, simulated annealing is a general form of optimization that mimics the quantum annealing process.

The maximum cut problem is a well-known example of a combinatorial optimization problem (Glover et al., 2019). Given a graph with vertices connected by edges, the goal of the maximum cut problem is to use one continuous cut to create as many partitions between the edges as possible (Commander, 2009). This problem has applications in fields like statistical physics (Commander, 2009).

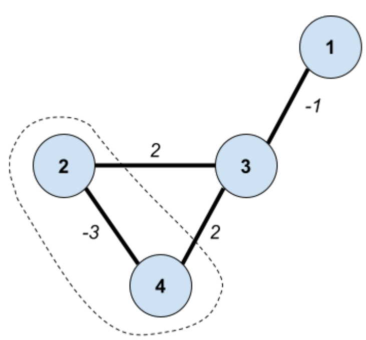

Figure 1: The maximum cut of a graph with weighted edges.

The above problem can be modeled using a QUBO formulation. First, the formulation to maximize the number of edges in the cut can be represented below.

$$\textit{Maximize}\ y = \sum_{(i,j)\in E} (x_i + x_j - 2x_i x_j)$$

Thus, the above graph can be formulated as the following, while accounting for the weights.

$$\textit{Maximize}\ y = -1(x_1+x_3-2x_1x_3)+2(x_2+x_3-2x_2x_3)-3(x_2+x_4-2x_2x_4)+2(x_3+x_4-2x_3x_4)$$

Further simplifying the above formulation yields the following.

$$\textit{Maximize}\ y = -x_1 - x_2 + 3x_3 - x_4 + 2x_1x_3 - 4x_2x_3 + 6x_2x_4 - 4x_3x_4$$

Finally, the formulation can be represented as a QUBO matrix.

$$\begin{bmatrix} -1 & 0 & 2 & 0 \\ 0 & -1 & -4 & 6 \\ 0 & 0 & 3 & -4 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Quantum computing and modeling Sudoku as an optimization problem presents itself as an effective way of solving even the most difficult Sudoku problems. Sudoku is a common number-based logic puzzle where players must fill in a 9x9 grid with digits so that no digit can appear multiple times in a row, column, or 3x3 subsquare. Sudoku puzzles can be solved using combinatorial optimization techniques (Lawler, 1985), which has garnered interest in developing mathematical expressions that represent the requirements of the puzzle in order to solve the Sudoku puzzle. Furthermore, Sudoku puzzles can be represented as graph coloring problems, where the goal is to create a 9-coloring of a graph when already given a partial 9-coloring (Lewis, 2016).

Graph coloring is a subset of graph theory, where the aim is to assign labels to elements of a graph, given certain constraints (Jensen and Toft, 2011). Graph coloring algorithms have many relevant applications in modern computer science such as data mining, image segmentation, clustering, and image capturing (Ahmed, 2012). Thus, developing new methods of solving Sudoku will advance graph coloring algorithms and their related fields like artificial intelligence, which requires large amounts of data and processing of that data to train neural networks.

Sudoku is an NP-hard problem, making it very difficult based on the initial conditions of the board (Machado and Chaimowicz, 2011). One current way of solving Sudoku is a brute-force approach known as backtracking, where each branch is completely explored before moving to another branch that could lead to the solution. While filling in the empty cells with a digit from available choices, if the algorithm finds that a digit violates the rules of Sudoku, it backtracks and changes the digit of the previous cell (Chatterjee et al., 2014). Another method of solving Sudoku is simulated annealing, which is a probabilistic optimization method. First, the empty cells are filled with random values, and the number of errors in the puzzle is counted. Then, the values of two cells are swapped and the number of errors is recounted. The algorithm will use the updated filled version as a starting point for future iterations if it has fewer errors than the original filled version (Chatterjee et al., 2014). However, there are issues that exist with these algorithms, such as the backtracking algorithm's slow solving time compared to other algorithms that use deductive methods. Thus, quantum computing can be applied to solve Sudoku problems by modeling them as optimization problems.

The goal of this project is to use Sudoku as a method of evaluating the performance of the latest quantum annealers. Another goal is to develop a new approach to formulate Sudoku problems and graph coloring problems, in general, as a QUBO formulation. It is hypothesized that this formulation will employ a novel method of reducing the number of variables in the formulation in order to use a quantum annealer to solve the problem. This formulation can then be applied to allow larger, more complex problems, like image segmentation and clustering, to be solved with quantum computing, thus further advancing the emerging field. This experiment was done by formulating a variety of Sudoku problems into QUBO formulations and revising those formulations in order to determine the limit of existing quantum hardware.

Methods and Materials

The main goal of this experiment was to determine if quantum computing can be used to solve Sudoku, a graph coloring problem, and to what extent quantum computing can solve it. This project was implemented using resources from D-Wave Systems.

In general, in order to solve problems on quantum computers, the problem must first be formulated as an objective function. Then, desired solutions can be found through sampling, the process of optimizing for low-energy states in the objective function.

Thus, the first step in this project was to formulate the Sudoku problem as an objective function. In Sudoku, there are a number of requirements that must be fulfilled in order to consider the puzzle complete. From these requirements, constraints can be derived in order to optimize the solution to the problem, which is solving the Sudoku puzzle. Let *i* represent the row, *j* represent the column, and *k* represent the digit of the hint.

1. Each cell can contain only a single number.

$$\sum_{k=1}^{9} X_{ijk} = 1, \forall ij \in cell$$

2. Each row in the whole puzzle must contain numbers 1-9 only once.

$$s.t. \sum_{j} X_{ijk} = 1, \forall i \in row, \forall k \in K(K = \{1...9\})$$

3. Each column in the whole puzzle must contain numbers 1-9 only once.

$$s.t. \sum_{i} X_{ijk} = 1, \forall j \in column, \forall k \in K(K = \{1...9\})$$

4. Each 3x3 subsquare in the whole puzzle must contain numbers 1-9 only once.

$$s.t. \sum_{j}^{2} \sum_{i}^{2} x_{(i+v)(j+v)k} = 1, \forall k \in K(K = \{1...9\})$$

$$u, v \in \{0, 3, 6\} : \textit{offset to each grid}$$

5. Hints cannot be altered.

$$s.t. \sum_{hint} X_{ijk} = 1$$

These constraints are then transformed into a QUBO formulation that the solver will attempt to minimize in order to reach the optimal solution.

Then, using starter code from D-Wave's library of tutorials, the objective function, along with the unsolved puzzle, was formatted as a binary quadratic model (BQM) for each problem.

The BQM was used as a precursor to the final QUBO model that would be sent to the solver. After the BQM was generated, it was converted to QUBO so that it could be reduced. The reduction was performed by removing variables where their values were already known. For example, if there was a variable $x_{179}x_{225}$, and it was already known that in row 1, column 7, the digit 9 could not be placed because it already appeared in row 1, then x_{179} could be rewritten as 0, thereby reducing $x_{179}x_{225}$ to 0 as well. Variable reduction accounted for the facts that hints cannot be changed and that rows, columns, and subsquares cannot contain the same digit.

Next, the process of quantum annealing was performed by submitting the reduced QUBO to the sampler. The sampler used the solver "DW_2000Q_6" which is D-Wave's quantum solver with 2041 working qubits, and the number of reads by the sampler was set to 1000. This is because finding the best solution is probabilistic, so a large number of samples are taken in order to ensure that a solution with the lowest possible energy is returned. The embedding was set to "EmbeddingComposite". Then, the solution with the lowest energy was verified by checking if there were duplicate digits in any row, column, or subsquare of the completed Sudoku problem. The metrics collected in quantum annealing were whether the puzzle was correctly solved using quantum annealing, the number of missing digits in the puzzle, the CPU runtime in seconds, the number of variables before reducing the QUBO formulation, the number of variables after reducing the QUBO formulation, the number of logical variables, the number of physical qubits used in embedding, and the minimum energy.

Simulated annealing was also performed for each problem in order to serve as a benchmark for performance. The sampler used was D-Wave's Simulated Annealing Sampler, and the number of reads was set to 1000 as well. Similarly to quantum annealing, the solution with the lowest energy was verified. The metrics collected for simulated annealing were the minimum energy and whether the puzzle was correctly solved using simulated annealing, but many metrics were shared between quantum annealing and simulated annealing, as those metrics were dependent on the Sudoku problem itself, like the number of missing digits in the puzzle, the number of variables before reducing the QUBO formulation, and the number of variables after reducing the QUBO formulation.

A variety of test cases were generated in order to accurately assess the performance of the annealers. Test cases were defined by the number of missing digits since more missing digits correlate to a harder Sudoku problem for the quantum computer. The number of missing digits for test cases ranged from 5 to 55 in intervals of 5. For each number of missing digits, 5 different test cases were generated. Thus, 55 test cases were generated. These test cases were created using a modified Sudoku solver and generator by Medium author Kush. This program generated Sudoku problems by populating a 9x9 board with random digits and removing digits at random while checking if the number of possible solutions to the new puzzle was exactly 1. Once the test cases were generated, they were run through both the quantum annealer and simulated annealer. Each of the 55 test cases was run 3 times through the quantum annealer, while also being run 5 times through the simulated annealer. The number of runs differed based on the annealer due to computational constraints and the amount of time it took for each problem to run through each annealer multiple times.

All programs were run on Google Colaboratory. The computer used during experimentation was a MacBook Pro on the operating system macOS Big Sur Version 11.7.2. Because this project was run solely on the computer, no safety precautions were needed or taken.

Results

The metrics collected and analyzed were the percentage of instances that the quantum annealer and simulated annealer were able to solve correctly, the number of missing digits in the puzzle, the CPU runtime in seconds, the number of variables before reducing the QUBO formulation, the number of variables after reducing the QUBO formulation, the number of logical variables, the number of physical qubits used in embedding, and the minimum energy.

The following table displays the average variable reduction for each number of digits removed from the Sudoku puzzle, such as all puzzles with 5 missing digits. Variable reduction was calculated by finding the average number of variables before reduction for each number of missing digits, subtracting the number of variables after reduction for each number of missing digits from that value, and dividing the entire value by the initial number of variables before reduction.

Missing digits	5	10	15	20	25	30
Avg variable reduction (%)	99.943	99.887	99.701	99.513	99.038	98.754
Missing digits	35	40	45	50	55	
Avg variable reduction (%)	97.536	96.321	93.889	90.094	86.707	

Figure 2: Average variable reduction (%) for each number of missing digits.

As seen in the table above, as the number of missing digits increases, fewer variables can be reduced, making the QUBO formulation bigger and more difficult for the quantum annealer to solve. Additionally, as the number of missing digits increases, the number of variables in the QUBO formulation before reduction increases linearly.

The following graph shows the average CPU runtime during quantum annealing in seconds for each number of missing digits. The line of best fit is an exponential curve with the equation $3.29e^{0.27x}$ and an R^2 value of 0.897, which suggests that the relationship is highly exponential. As the number of missing digits increases, the average CPU runtime increases exponentially because there are more variables in the QUBO formulation, which demands more computational resources and calculations to solve.

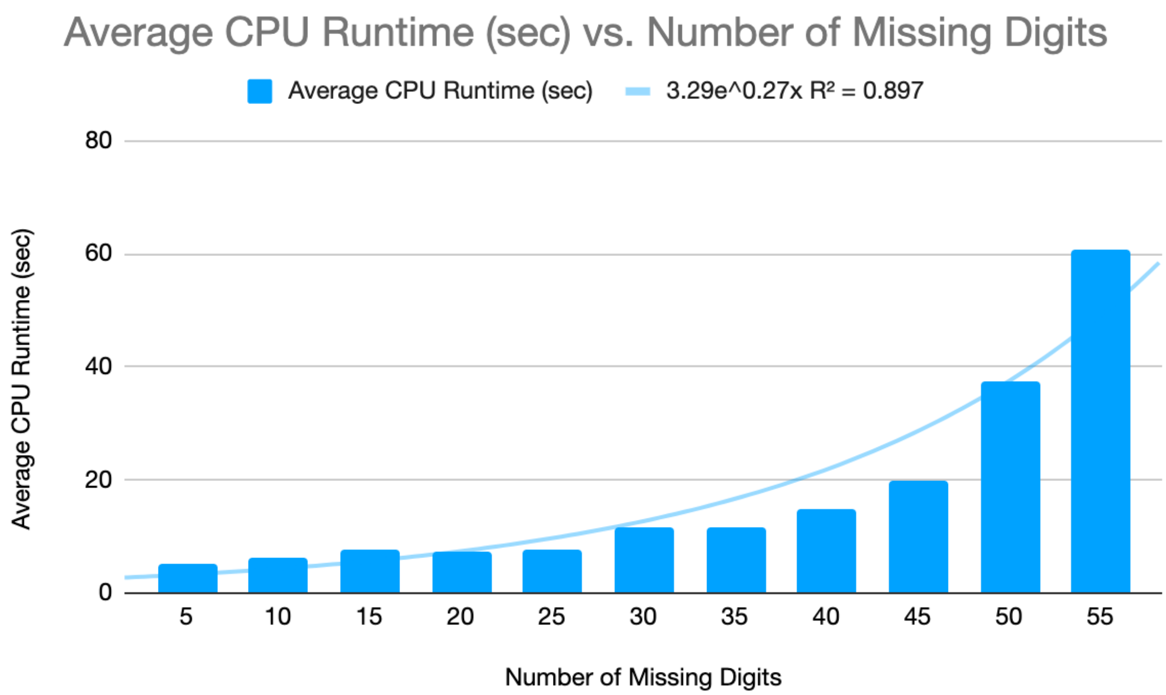

Figure 3: Average CPU runtime during quantum annealing for each number of missing digits and line of best fit.

For all puzzles with the number of digits removed ranging from 5 to 40, the quantum annealer was able to solve all instances. However, when attempting to solve puzzles with 45 digits missing, it was only able to correctly solve 33.33% of them, and for 50 and 55 digits missing, it was not able to correctly solve any of them. The quantum annealer was able to solve all "special case" Sudoku instances of 4x4 and 6x6, no matter the number of missing digits in the puzzle. The simulated annealer performed slightly better, solving all puzzles with the number of missing digits ranging from 5 to 45. It was able to solve 92% of Sudoku puzzles with 50 digits missing and 16% of puzzles with 55 digits missing.

The following graph displays the average number of logical variables and physical qubits used in embedding for each number of digits removed from the Sudoku puzzle. Both the average number of logical variables and the average number of physical qubits have an exponential relationship with the number of missing digits. The line of best fit for the relationship between the number of missing digits and the number of logical variables is $10.1e^{0.308x}$, and it had an R^2 value of 0.992. The relationship between the number of missing digits and the number of physical qubits can also be represented exponentially, with the equation $3.39e^{0.611x}$, which has an R^2 value of 0.994. Both of these extremely high R^2 values suggest that the relationships are highly exponential.

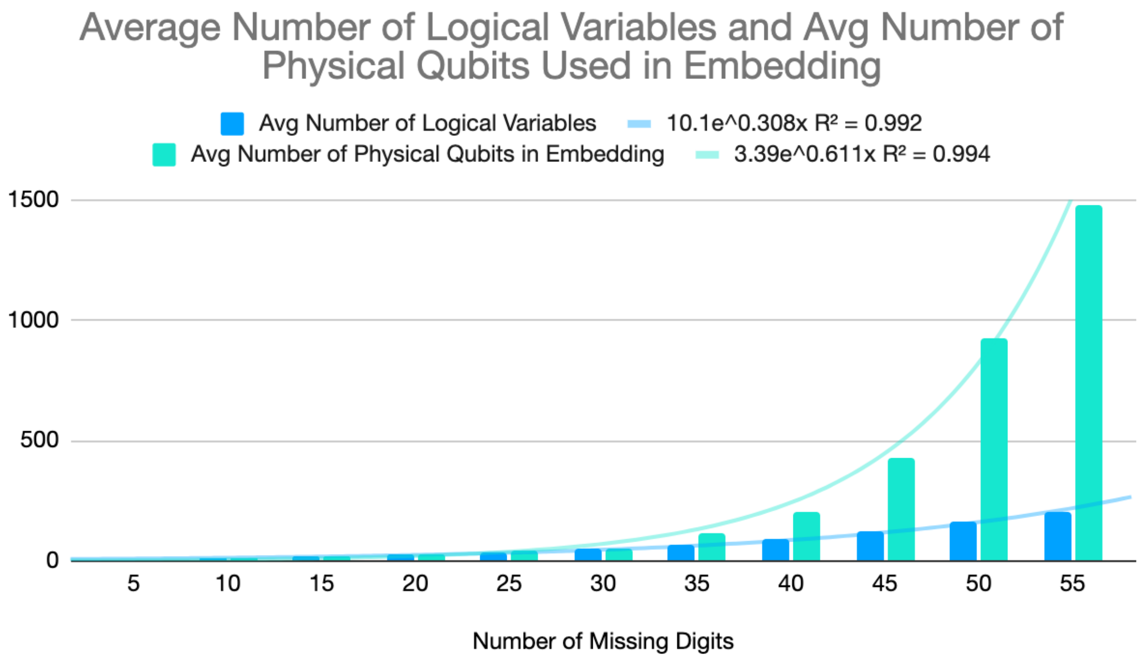

Figure 4: Average number of logical variables and physical qubits used in embedding for each number of missing digits and corresponding lines of best fit.

The average minimum energy for each number of missing digits varied for quantum annealing and simulated annealing. For quantum annealing, the average minimum energy began at -20 for all test cases with 5 digits removed. This minimum energy decreased at a constant rate of -4 for each additional digit removed until 40 missing digits. For example, all test cases with 10 digits removed had a minimum energy of -40. In fact, each test case for each number of missing digits had the exact same minimum energy. This pattern continued until 40 missing digits. However, at 45 missing digits, the average minimum energy was -173.87 and started to plateau with an average minimum energy of -175.0666667 for 50 missing digits and -175.2 for 55 missing digits. Simulated annealing had a different pattern for minimum energy. The minimum energy for every single test case with the number of missing digits ranging from 5 to 45 was 0.

For 50 missing digits, the average minimum energy was 0.32, and for 55 missing digits, the average minimum energy was 3.36.

Discussion and Conclusions

The results from this experiment show that quantum annealing can be used to solve Sudoku problems of various sizes and difficulties. However, there was a general plateau in the abilities of both quantum annealing and simulated annealing to solve Sudoku problems with fewer initial hints.

The average variable reduction percentages show that the method of reducing variables discussed previously was effective because it reduced over 85 to 99% of all variables. This reduction allowed the quantum annealer to solve Sudoku problems that it previously couldn't, showing it was effective. However, the reduction method did not allow every single test case to be solved, especially the test cases with more missing digits, and thus much larger numbers of variables in the QUBO formulation. A goal of future work would be to refine the variable reduction methodology in order to further reduce the number of variables in the QUBO formulation. Simulated annealing slightly outperformed quantum annealing in solving Sudoku puzzles with 45 or more missing digits correctly. This is likely because the number of variables in the QUBO formulation was too much for the quantum annealer to properly solve, thus suggesting a need to further improve the variable reduction technique. The patterns observed in the minimum energies for simulated annealing were expected because the annealer optimized for the lowest possible energy. Therefore, the correct solutions should have the lowest possible energy of zero. This can be seen in instances with the number of missing digits ranging from 5 to 45. Conversely, the instances that the simulated annealer was not able to solve did not have a minimum energy of 0, but instead, it was a positive value.

In addition to improving the method of reducing the number of variables in the QUBO formulation, the performance of different D-Wave annealers could be explored. D-Wave offers many other quantum annealers, like "Advantage_system6.1", which has 5616 working qubits. However, there are constraints to this exploration as current free access to D-Wave's annealers is the "Developer" access.

In this paper, the performance of a current state-of-the-art quantum annealer was benchmarked using Sudoku, a type of graph-coloring problem. This was done by creating

Sudoku problems of various sizes and difficulties and running them through both a quantum annealer and a simulated annealer. The performances were compared, and it revealed that quantum annealing can be used to solve most cases of Sudoku, although it does have some limitations. An exponential relationship between the number of missing digits in the Sudoku puzzle and other variables, like CPU runtime, the number of logical variables, and the number of physical qubits used in embedding, was revealed as well. This paper attempted to address the current gap in knowledge regarding how well quantum annealing can solve certain optimization problems, like Sudoku, as the field of quantum computing is still emerging. Although Sudoku has been applied to quantum computing before, hybrid solvers were used to solve the problems (Pal et al., 2020). This project is different in the sense that a pure quantum annealer was used, instead of a hybrid solver that combines both classical and quantum elements. Furthermore, the novel variable reduction technique applied to the QUBO formulations in this project can be applied to larger, more complex problems in the field of graph-coloring, like data mining, clustering, and image segmentation and capturing, thereby advancing the applications of quantum computing.

Literature Cited

Peer-reviewed and Scholarly References

Ahmed, S. (2012). Applications of graph coloring in modern computer science. International Journal of Computer and Information Technology, 3(2):1–7.

Chatterjee, S., Paladhi, S., and Chakraborty, R. A comparative study on the performance characteristicsof sudoku solving algorithms. IOSR Journals (IOSR Journal of Computer Engineering), 1(16):69–77.

Commander, C. W. (2009). Maximum cut problem, max-cut. Encyclopedia of Optimization, 2.

Glover, F., Kochenberger, G., and Du, Y. (2019). Quantum bridge analytics i: a tutorial on formulating and using qubo models. 4OR, 17:335–371.

Jensen, T. R. and Toft, B. (2011). Graph coloring problems. John Wiley & Sons.

Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lu, Z., Wang, ¨ H., and Wang, Y. (2014). The unconstrained binary quadratic programming problem: a survey. Journal of combinatorial optimization, 28:58–81.

Lawler, E. L. (1985). The traveling salesman problem: a guided tour of combinatorial optimization. Wiley-Interscience Series in Discrete Mathematics.

Lewis, R.M.R. (2016). Introduction to Graph Colouring. In: A Guide to Graph Colouring. Springer, Cham. https://doi.org/10.1007/978-3-319-25730-3_1

Li, Y., Tian, M., Liu, G., Peng, C., and Jiao, L. (2020). Quantum optimization and quantum learning: A survey. Ieee Access, 8:23568–23593. 13

Machado, M. C. and Chaimowicz, L. (2011). Combining metaheuristics and csp algorithms to solve sudoku. In 2011 Brazilian Symposium on Games and Digital Entertainment, pages 124–131. IEEE.

Morita, S. and Nishimori, H. (2008). Mathematical foundation of quantum annealing. Journal of Mathematical Physics, 49(12):125210.

National Academies of Sciences, E., Medicine, et al. (2019). Quantum computing: progress and prospects.

Pal, A., Chandra, S., Mongia, V., Behera, B. K., and Panigrahi, P. K. (2020). Solving sudoku game using a hybrid classical-quantum algorithm. Europhysics Letters, 128(4):40007

Preskill, J. (2018). Quantum computing in the nisq era and beyond. Quantum, 2:79