Optimizing Quantum Annealing to Advance Graph Coloring Algorithms

Anooshka Pendyal

Deep Run High School, Glen Allen, Virginia, United States of America

Introduction

- **Quantum annealing (QA)** is a new approach to solve **optimization problems**, leveraging quantum mechanic effects such as quantum superposition and quantum tunneling. In QA, solving multivariable optimization problems is framed as finding ground state energy of spin configurations.

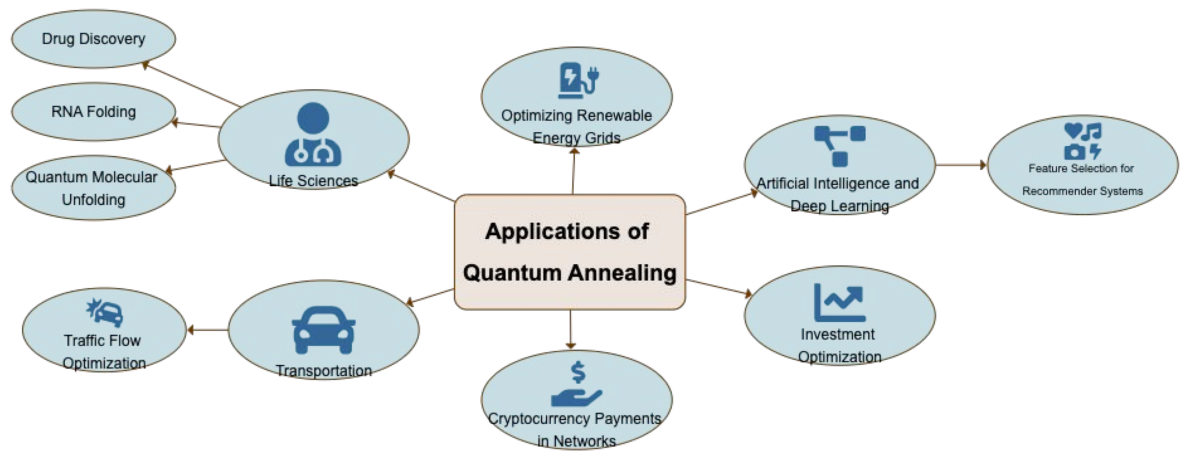

Figure 1: Industry applications of quantum annealing. Quantum annealing has a very wide range of uses, from drug discovery to environmental sustainability. Figure created by Anooshka Pendyal.

- **Sudoku** puzzles can be solved using **combinatorial optimization** techniques and can be represented as **graph coloring problems**, where the goal is to create a 9-coloring of a graph.

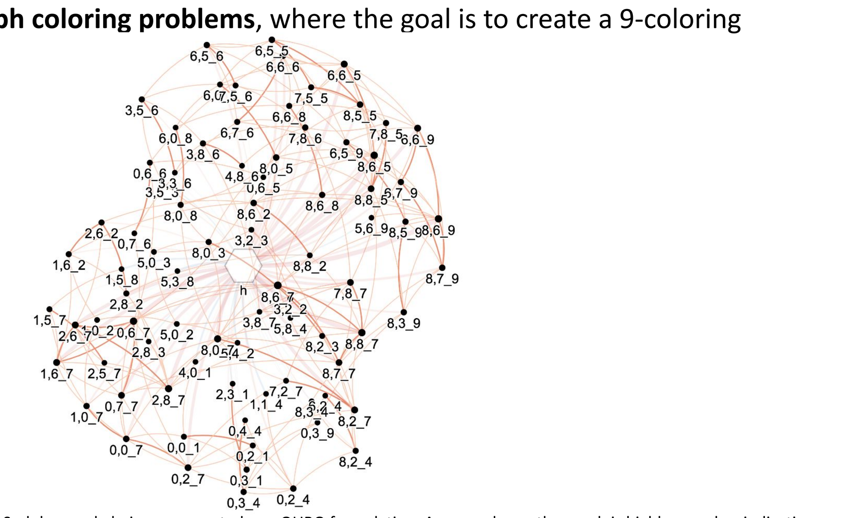

Figure 2: A graphical depiction of a Sudoku puzzle being represented as a QUBO formulation. As seen above, the graph is highly complex, indicating that the QUBO formulation will be large and difficult for the quantum annealer to solve. This presents a need to reduce the number of variables in the QUBO forumulation. Figure created by Anooshka Pendyal using the HamGraph library.

- **Graph coloring algorithms** aim to assign labels to elements of a graph, given certain constraints, and have many relevant applications in modern computer science such as **data mining**, image segmentation, clustering, and image capturing. Thus, developing new methods of solving Sudoku will **advance** graph coloring algorithms and their related fields like **artificial intelligence and networking**.
- Current methods of solving Sudoku, like backtracking, have slow solving times, indicating that quantum computing may provide an advantage by modeling Sudoku problems, which are NP-hard, as optimization problems.
- Sudoku instances have been solved on **hybrid CPU-QPU** in the past but not with **pure QA**.
- Solving Sudoku using only quantum devices is **limited by the hardware** and **the problem sizes.**

Objectives

1. Develop an **efficient formulation** for Sudoku and graph-coloring problems on D-Wave quantum annealers as a QUBO formulation.
   a. **Reduce the formulation size** (number of variables and quadratic terms) through a **novel variable reduction technique**, which can 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.
   b. **Increase the probability** of finding solutions
2. **Benchmark the performance** of quantum annealers on Sudoku instances of different difficulty levels

Optimization on Quantum Annealers

- **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**. They have a wide range of applications from artificial intelligence to finance and can be used to help answer questions like "How can asset managers find the optimal portfolio of assets, given the basket of options, a client's budget, and risk-appetite?"

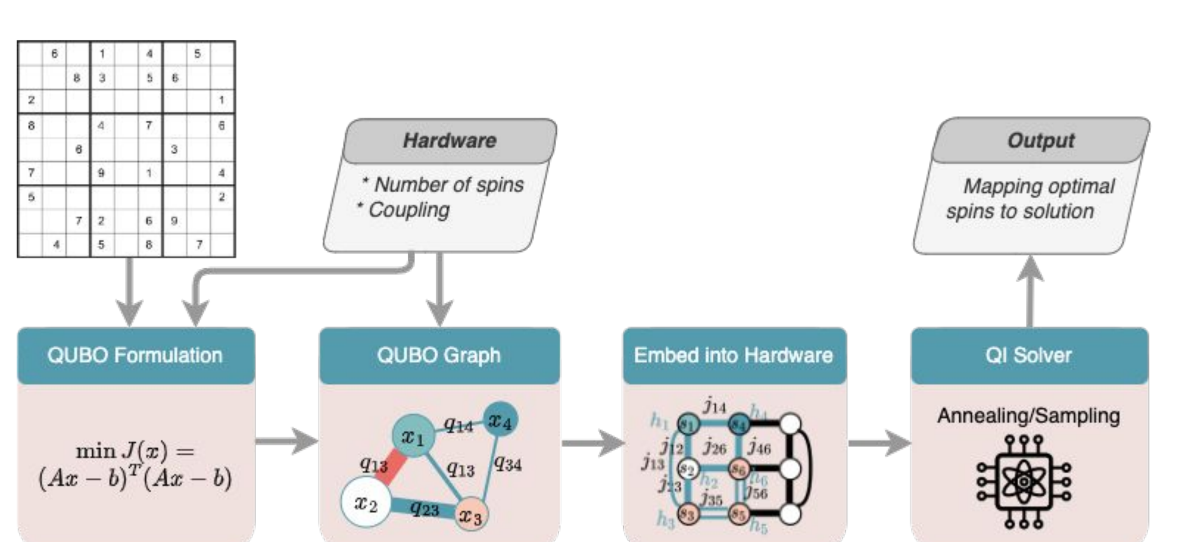

Figure 3: Depiction of a typical quantum algorithm workflow on a D-Wave quantum annealer. Based on a combinatorial optimization problem, a quadratic unconstrained binary optimization (QUBO) formulation is developed, which is embedded and submitted to the annealer. It is solved by searching for minimum energy solutions. Figure created by Anooshka Pendyal.

Methodology

Efficient QUBO Formulation for Sudoku

1) **Formulate Sudoku requirements** into constraints in the QUBO formulation. 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.
   $$s.t. \sum_{j}^{2} \sum_{i}^{2} x_{(i+v)(j+v)k} = 1, \forall k \in K(K = \{1...9\})$$
   5. Hints cannot be altered.
   $$s.t. \sum_{hint} X_{ijk} = 1$$

The final objective function is set to H, the minimum combinatorial value.

$$H = \alpha \sum_{ij} (\sum_{k=1}^{9} x_{ijk} - 1)^2 + \alpha \sum_{k=1}^{9} \sum_{i} (\sum_{j} x_{ijk} - 1)^2 + \alpha \sum_{k=1}^{9} \sum_{j} (\sum_{i} x_{ijk} - 1)^2 + \alpha \sum_{k=1}^{9} (\sum_{j}^{2} \sum_{i}^{2} x_{(i+v)(j+v)k} - 1)^2 + 2\alpha \sum_{hint} (1 - x_{ijk})$$

2) **2x2 Sudoku Example without variable reduction.** In this example, the formulation has been slightly modified to include values of 1 and 2 and not 1 through 9, as this Sudoku puzzle only has values of 1 and 2.

1	
2	1

$$H = \alpha \sum_{ij} (\sum_{k=1}^{2} x_{ijk} - 1)^2 + \alpha \sum_{k=1}^{2} \sum_{i} (\sum_{j} x_{ijk} - 1)^2 + \alpha \sum_{k=1}^{2} \sum_{j} (\sum_{i} x_{ijk} - 1)^2 + 2\alpha \sum_{hint} (1 - x_{ijk})$$

The objective function for the 2x2 Sudoku puzzle, with α set to an arbitrary value of 2, is expanded. Constants are disregarded.

$$-10x_{111}^2 - 6x_{112}^2 - 6x_{121}^2 - 6x_{122}^2 - 6x_{211}^2 - 10x_{212}^2 - 10x_{221}^2 - 6x_{222}^2 + 4x_{111}x_{112} + 4x_{121}x_{122} + 4x_{211}x_{212} + 4x_{221}x_{222} + 4x_{111}x_{211} + 4x_{121}x_{221} + 4x_{112}x_{212} + 4x_{122}x_{222} + 4x_{111}x_{121} + 4x_{211}x_{221} + 4x_{112}x_{122} + 4x_{212}x_{222}$$

Finally, the formulation is transformed into the QUBO matrix by placing the coefficient of each variable into the matrix. As seen below, the QUBO matrix is unnecessarily big for a simple puzzle, indicating the need for variable reduction.

$$\begin{bmatrix} -10 & 4 & 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & -6 & 0 & 4 & 0 & 4 & 0 & 0 \\ 0 & 0 & -6 & 4 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & -6 & 4 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 \end{bmatrix}$$

1) Formulate Sudoku problem as objective function

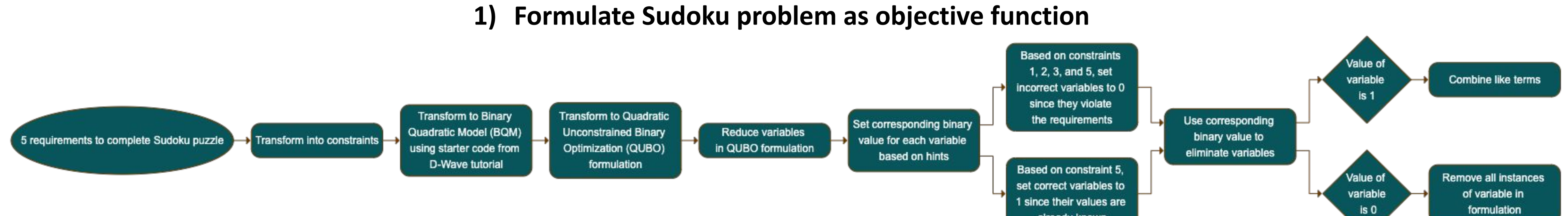

Figure 4: A flowchart depicting the algorithm implemented to formulate the Sudoku problem as an objective function. Figure created by Anooshka Pendyal.

2) Sampling in quantum annealing

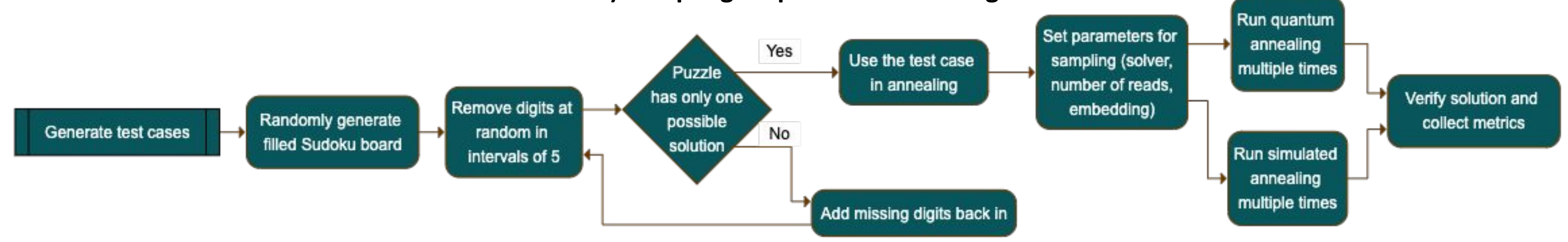

Figure 5: A flowchart depicting the algorithm implemented to generate and solve test cases with both quantum annealing and simulated annealing. Figure created by Anooshka Pendyal.

Results

Quantum annealing

- Able to **solve all puzzles with the number of digits removed ranging from 5 to 40**
- Could only solve **60% of puzzles with 45 missing digits** and could not solve any puzzles with more than 50 digits missing
- Able to solve **all "special case"** Sudoku instances (ex. 4x4 and 6x6) independent of the number of missing digits

Simulated annealing

- **Performed slightly better** in solving all puzzles with the number of missing digits ranging from 5 to 45
- Able to solve 92% of puzzles with 50 digits missing and 16% of puzzles with 55 digits missing

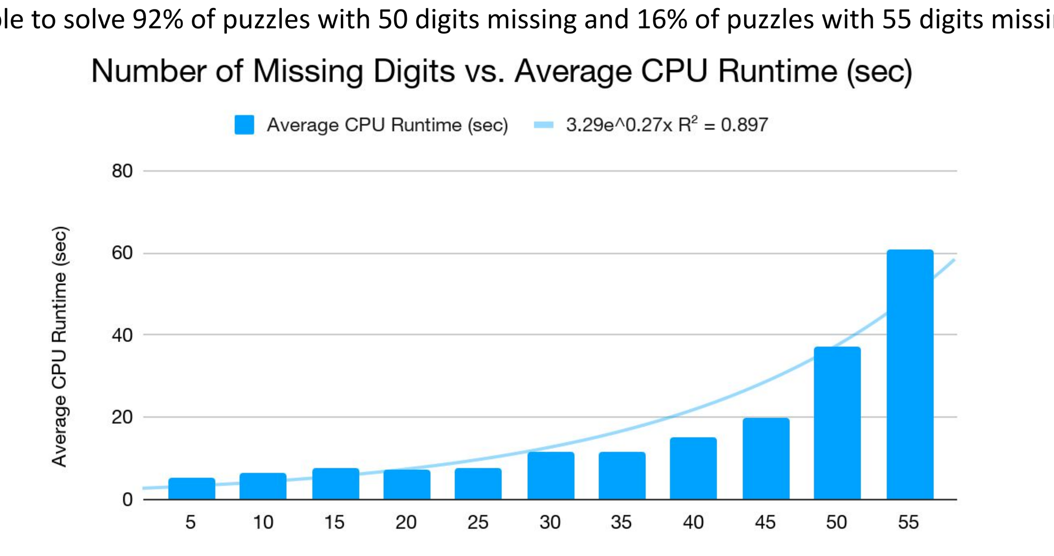

Figure 7: Average CPU runtime during quantum annealing for each number of missing digits and line of best fit. As the number of missing digits increases, the average CPU runtime increases exponentially because there are more variables in the QUBO formulation, which demands more computational resources and calculations to solve. Figure created by Anooshka Pendyal.

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 6: Average variable reduction (%) for each number of 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. As seen in the table, 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. Figure created by Anooshka Pendyal.

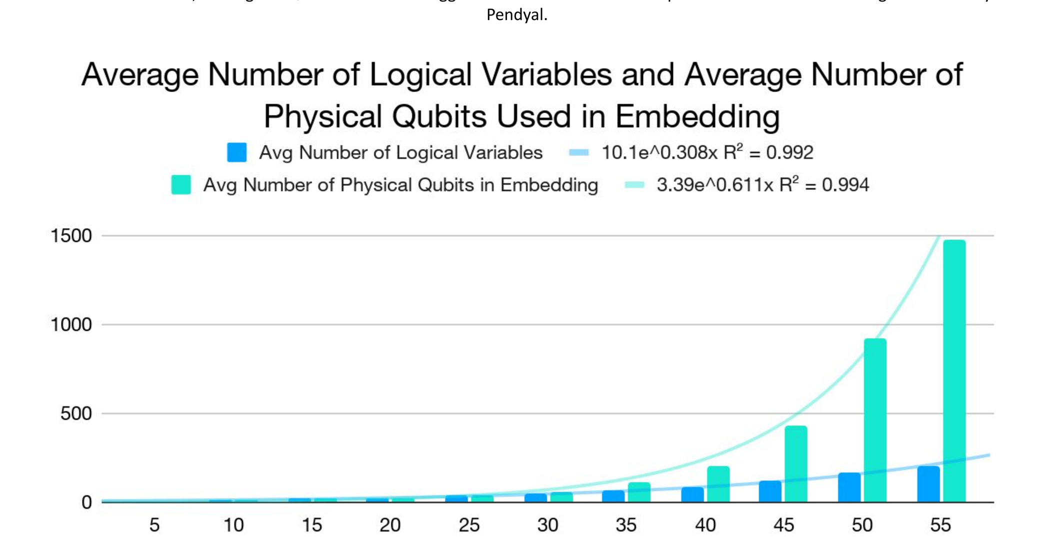

Figure 8: Average number of logical variables and physical qubits used in embedding for each number of missing digits and corresponding lines of best fit. Both the average number of logical variables and the average number of physical qubits have an exponential relationship with the number of missing digits, which is supported by the extremely high R^2 values. Figure created by Anooshka Pendyal.

Discussion and Conclusions

- **Quantum annealing can be used to solve Sudoku problems** of various sizes and difficulties, but there are still limitations.
- The **novel variable reduction method was highly effective** because it reduced over 85 to 99% of all variables and allowed the quantum annealer to solve Sudoku problems that it previously could not.
- These limitations in both quantum and simulated annealing were likely a result of the number of variables in the QUBO formulation being too much for the annealers to properly solve.
- Thus, for future work, a goal would be to **further refine the variable reduction methodology through a new embedding technique**.
- The performance of different D-Wave annealers could be explored as well.
- This project **addresses a current gap in knowledge** because 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**.

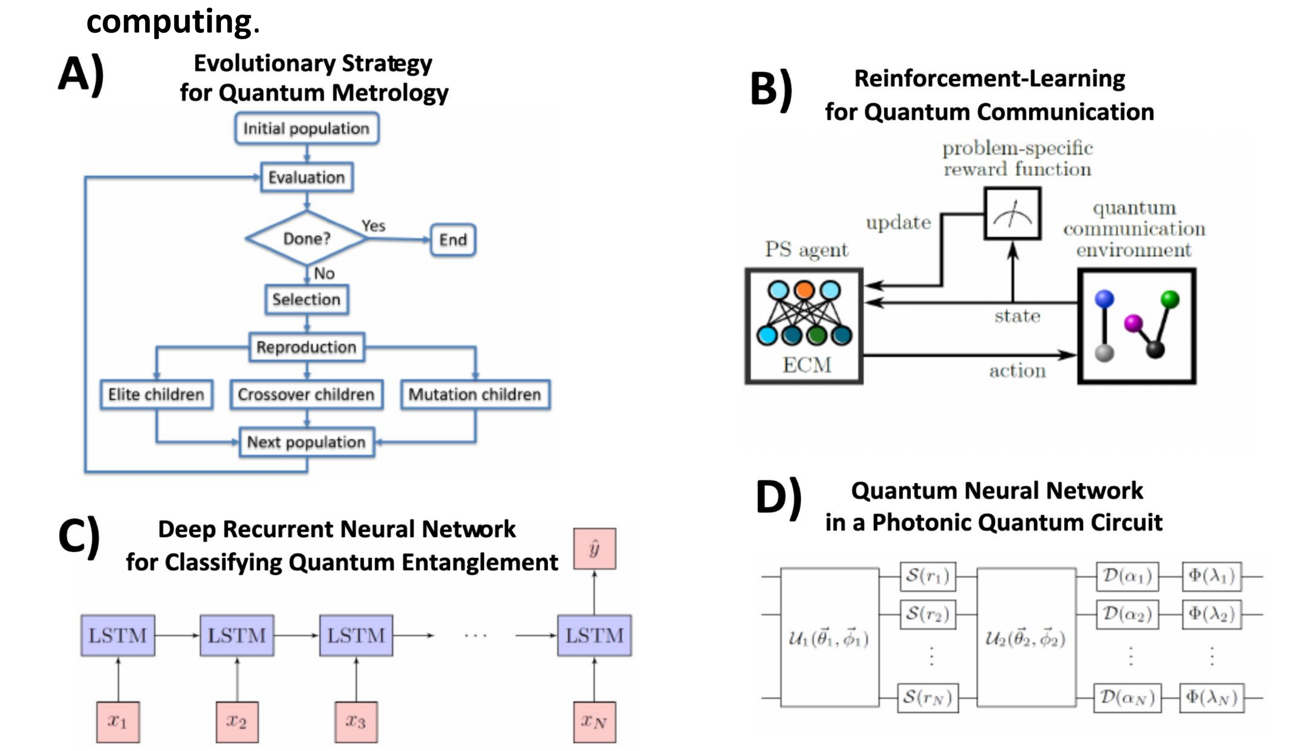

Figure 9: Class of algorithms for computer-inspired quantum experiments with various optimization and machine learning techniques. Adapted from "Computer-inspired Quantum Experiments". A) An evolutionary approach to design novel methods in quantum metrology. A genetic algorithm is used to make the initial random population undergo an evolutionary process that selects the best individuals based on their fitness and mutates them to form the next generation. B) Using a reinforcement learning algorithm to design quantum communication schemes. An agent performs an action in changing the quantum communication scheme, thereby changing the state of the environment and receiving a reward based on the quality of the action. C) A deep recurrent neural network learns to predict quantum entanglement properties. This process approximates the time-consuming objective function with a fast neural network, which has the potential to speed up the search for new quantum entanglement experiments significantly. D) A photonic quantum circuit follows the quantum neural network Ansatz, and the circuit can be parametrized continuously, allowing for gradient-based optimization techniques.

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 characteristics of 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.

Krenn, M., Erhard, M., & Zeilinger, A. (2020). Computer-inspired quantum experiments. Nature Reviews Physics, 2(11), 649-661.

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

Acknowledgements

A special thank you to Dr. Thang Dinh and Phuc Thai at VCU's Department of Computer Science for providing support and guidance on this project!