USING QUANTUM ANNEALING FOR ROUTING AND WAVELENGTH ALLOCATION OPTIMIZATION IN OPTICAL NETWORKS

by

VICTOR VIRAG

SN: 23215220

MSC QUANTUM TECHNOLOGIES

August 2024

Department of Physics and Astronomy

University College London

Supervised by: Dr. Alejandra Beghelli

Contents

Abstract

This work benchmarks Quantum Annealing (QA), Hybrid Annealing (HA), and Simulated Annealing (SA) for solving Wavelength Allocation (WA) and Routing and Wavelength Allocation (RWA) problems in static Wavelength Division Multiplexing (WDM) optical networks. We introduce a novel Quadratic Unconstrained Binary Optimization (QUBO) formulation for the static RWA problem, designed to minimize link utilization, distinguishing it from existing dynamic formulations [1]. Our approach is effective for small networks but is limited by exponential scaling. We also compare solving the WA problem on quantum and quantum-inspired solvers against a greedy heuristic and the Gurobi solver. Although Gurobi guarantees global optimality, its run time becomes impractical for larger problems. In our benchmarks, the greedy heuristic not only provided optimal solutions in all cases where the exact solutions were known, but also did so with computation times measured in milliseconds, significantly outperforming all other approaches. Quantum and quantum-inspired approaches show some potential but are limited by scalability issues. When solving WA. on average, HA allocated 7.1% more wavelengths than the greedy LDF, QA allocated 5.6% more, and SA allocated 41.7% more. The project's source code is available under the MIT licence, with links in the Appendix (8.2). This work builds on Boev et al. (2023), who provided QUBO formulations of WA for quantum-inspired algorithms [2].

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Alejandra Beghelli, for her invaluable advice and support throughout this project. I greatly appreciate the time she dedicated to helping me and her commitment to ensuring that I achieved a high standard in this work.

I am also thankful to Sergey Usmanov for responding to my inquiries and providing valuable insights related to their work on quantum-inspired optimization for WA [2].

I would also like to express my heartfelt thanks to my parents, Gergana and Csaba, and my brother, Robin, for their continuous support. A special thanks to my partner, Lora, for her patience and encouragement. I am also grateful to my friends Simon, Sara, Vasko, Lorinc, and Martin for their advice and support. Lastly, I thank The Basilisk for its grace and mercy.

1 Introduction

The growing demand for high-speed internet has continually driven advancements in optical networks, with Wavelength Division Multiplexing (WDM) [3] being a foundational technology in the field. WDM significantly increases data transmission capacity by allowing multiple signals to travel through the same optical fibre on different wavelengths. However, efficiently managing network routing and wavelength allocation for large networks remains computationally difficult, and is described by the well-studied problems of Wavelength Allocation (WA) [4] and Routing and Wavelength Allocation (RWA) [5].

This work explores the use of Quantum Annealing (QA), Hybrid Annealing (HA) and Simulated Annealing (SA) to solve WA and RWA problems in optical networks. We focus on Wavelength-Selective (WS) networks using Fibre Switch Cross-Connects (FXC) [6] or Wavelength-Selective Cross-Connects (WSXC) [6], where the allocated wavelength remains consistent across the entire path. Our work specifically addresses only WS networks, leaving out considerations of other network types.

A key topic of this work is using the Quadratic Unconstrained Binary Optimization (QUBO) formulation for solving Combinatorial Optimization (CO) problems. We compare the performance of QUBO-based solutions on D-Wave quantum hardware and simulated annealers with classical solvers, such as the greedy Largest Degree First (LDF) heuristic [7], and the Gurobi industrial combinatorial solver.

We also introduce a novel QUBO formulation for solving the static RWA problem. We demonstrate this approach on a small toy network of 6 nodes and 20 links and analyse its scalability challenges. We also solve the WA problem for real-world network topologies, including Eurocore, EON, ARPANET, UKnet, Eurolarge, and USnet, using routing configurations generated through uniform random sampling. The solvers used include Gurobi, greedy LDF heuristic, SA, HA, and QA.

For the reader's convenience, the subsections below offer brief introductions to the core topics of this work, with more detailed explanations and mathematical formulations provided in later sections.

1.1 RWA and WA Problems

In optical networks, WDM is used to transmit multiple signals through an optical fibre cable at different wavelengths of light with minimal signal interference. In WS optical networks, a lightpath is a route using a single wavelength from the source node to the destination node. The WA problem is a CO problem that attempts to allocate the minimal number of wavelengths to all routes in a network, ensuring that no two routes sharing a link use the same wavelength [4]. The process of selecting the links for the route is known as the routing problem [8]. In RWA, the objective is to solve both the routing and WA problems simultaneously, aiming to minimize the total number of wavelengths used.

The type of routing used, static or dynamic, determines whether the network can adapt in real-time to changes in the topology [9]. Static routing solutions are computed offline and can include precomputed alternate paths [10]. This work uses statically precomputed routes for solving RWA. Dynamic routing, on the other hand, depends on the current network state and allocates optimal routes as needed, allowing the network to adapt to changes such as link failures in real-time [10, 9].

1.2 The Ising Model and Hamiltonians

The Ising model is a theoretical representation of interacting molecular spins in a lattice structure, influenced by an external magnetic field [11]. It serves as a model of electromagnetic dynamics in a spin system with the primary goal of determining the system's lowest energy configuration, referred to as the ground state. A Hamiltonian, which is a mathematical representation of the system's total energy and is fundamental in both classical and quantum mechanics, describes these interactions. In Quantum Computing (QC), problems formulated using the Ising model can be mapped onto the qubit states of QA systems through an embedding process. These systems evolve over time under the principles of quantum thermodynamics [12, 13], driving the system towards minimizing its Hamiltonian and finding the optimal solution.

1.3 QUBO Overview

QUBO models provide a framework for formulating many CO problems in a form which can be solved using problem-independent methods. Formulations exist for numerous graph theory problems, such as the Graph Colouring Problem (GCP) [14, 15], Max-Cut [15], and Multiple Knapsack [15]. The significance of QUBO problems in QC is due to their equivalence to the Ising problem [16], allowing them to be transformed into Ising Hamiltonians that can be mapped onto quantum systems with relatively low overhead [17]. The system's Hamiltonian can then be evolved to its minimal energy state, which corresponds to the optimal solution. While much of the focus for solving QUBO has been on using quantum-inspired algorithms [18, 19] and QA systems [17], the formulation also shows potential for use in neuromorphic computing architectures [20].

QUBO problems have shown potential in addressing real-world issues in economics [21], optical networks [2], and machine learning [22], using both quantum [17] and quantum-inspired [18] solvers.

1.4 Quantum Computing

Quantum computers use the phenomena of entanglement and superposition to gain better computational capabilities compared to classical computers. They currently face a significant technological challenge with decoherence, the process by which quantum information is lost to the environment [23], which degrades the quality of computations. Despite this, some existing quantum systems can perform useful computations using noisy algorithms that tolerate a certain level of decoherence. Currently, QC is in the Noisy Intermediate-Scale Quantum (NISQ) era, where devices are able to solve some non-trivial optimization problems.

There are several hardware approaches to QC, including superconducting quantum chips [24], ion-trap devices [25], photonic chips [26], and quantum annealers. This work focuses on the analogue approach known as QA, also referred to as adiabatic quantum computation, notably adopted by D-Wave [27]. QA solves optimization problems by encoding them as a Hamiltonian and evolving the system to its ground state, which represents the optimal solution [17]. Due to the probabilistic nature of quantum mechanics, computations are rerun multiple times to converge to a solution.

2 QUBO

2.1 QUBO Definition and Problem Formulation

We provide a formal definition of the QUBO problem, as presented in [15, 28].

Let $x_1, \ldots, x_n$ be binary variables defined by the CO problem, such that $x_1, \ldots, x_n \in \{0, 1\}$. Let Q be a symmetric square matrix of size $n \times n$ with integer elements, such that $\forall i, j$, $Q_{ij} \in \mathbb{Z}$. The elements of Q encode the magnitude of interactions between the binary variables x_i, x_j, and form the problem's objective function. The quadratic component of QUBO comes in as the quadratic terms in the objective function, which can be written down as such:

$$\sum_{ij} Q_{ij} x_i x_j \tag{1}$$

QUBO problems are unconstrained, meaning there are no explicit constraints encoded in the problem structure. As a result, QUBO solvers can produce solutions that may not be valid when the underlying problem has strict requirements that must be adhered to. To fix this, solution verification can be used to rule out invalid results. To reduce the probability of outputting sub-optimal solutions, the parameters of the objective function encoded in the matrix Q are often adjusted to penalize low-quality results [15].

The symmetric QUBO matrix Q is constructed as follows [15]:

$$Q_{ij} = \begin{cases} q_{ij} & \text{if } i = j, \\ \frac{q_{ij}+q_{ji}}{2} & \text{if } i \neq j. \end{cases} \tag{2}$$

The goal of QUBO problems is to find a $n \times 1$ binary solution vector s, which minimizes the value y in the equation [15]:

$$y = s^T Q s \tag{3}$$

2.2 Solving QUBO Problems

Common approaches to solving QUBO problems include classical and quantum-inspired algorithms such as exact solvers [29, 30], heuristic methods [31, 32], and metaheuristic techniques [33].

Exact solvers, such as GLPK and Gurobi, solve Integer Linear Programming (ILP) and Mixed-Integer Linear Programming (MILP) models, which describe the optimization problem in terms of an objective function and constraints. While these solvers provide the best possible solutions, they can be computationally intensive and impractical for large problem instances.

Heuristic methods, such as greedy algorithms and local search techniques, offer significantly faster execution time at the cost of not guaranteeing an optimal solution. Metaheuristic techniques, such as genetic algorithms [34] and tabu search [35], explore the solution space more thoroughly and aim to improve solution quality while maintaining a speed that is comparable to heuristic approaches.

Quantum-inspired methods emulate the behaviour of quantum systems but run on classical hardware. These methods are claimed to be effective for large-scale optimization problems [2, 18]. Notable approaches include simulators of the Coherent Ising Machine (CIM) [18] and QA [36].

When using quantum solvers, a QUBO problem can be transformed into an Ising problem and solved on quantum annealers [17, 37] and Coherent Ising Machines (CIMs) [38]. The quantum nature of CIMs has been debated, with some researchers arguing that their fast performance stems from quantum effects, such as optical squeezing [39], while others believe the speedup is due to classical analogue optical processing [18]. We reference this debate to highlight the ongoing uncertainty about the quantum contributions of CIMs. For this reason, in this work, we focus exclusively on QA approaches for solving QUBO with quantum hardware, where the quantum contributions are more well-established [40, 41].

2.3 Research and Applications of QUBO

QUBO problems have already been applied to real-world challenges in both research and industry, with D-Wave being a prominent example of industrial application using quantum hardware. D-Wave has developed quantum annealers specifically designed for medium and large-sized QUBO problems [42, 27]. However, details about the specific problems solved on D-Wave systems are limited due to proprietary concerns.

In the research domain, QUBO formulations have been applied to a wide range of problems. For example, in machine learning [22], QUBO is used to transform training tasks for models such as linear regression, support vector machines, and balanced k-means clustering into a format compatible with quantum annealers. In genome assembly [43], QUBO is used to solve tasks associated with de novo genome assembly. Nuclear reactor fuel management [44] could also benefit from QUBO's ability to optimize the arrangement of fuel rods to improve safety and efficiency.

Additionally, WA in optical networks, which is a key aspect of this work, can be formulated in QUBO form and solved to optimize the assignment of available wavelengths [2].

3 The Ising Problem

3.1 Formulation of the Ising Model

The Ising model is a theoretical framework depicting coupled molecular spins on a lattice that may also be influenced by an external magnetic field [11]. Spins σ_i are classical and can be in one of two states: +1 (up) or -1 (down). The primary goal of the Ising model is to describe the interactions between these spins.

$$J_{ij} = \begin{cases} 1 & \text{if spin } i \text{ is a nearest neighbor to spin } j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\mu \sum_i H_i \sigma_i \tag{5}$$

$$E = H(\sigma) = -\sum_{ij} J_{ij} \sigma_i \sigma_j + \mu \sum_i H_i \sigma_i \tag{6}$$

The interaction between i and j is often characterized by a symmetric coupling matrix J, where J_{ij} represents the interaction strength between the spins i, j. The construction of matrix J for a 2D lattice problem is described in Equation (4) [11]. The effect of an external magnetic field H (not to be mistaken with the Ising Hamiltonian $H(\sigma)$) acting on each spin can also be included, as seen in Equation (5) [11]. Here, H_i denotes the magnetic field component influencing spin i, and μ represents the magnetic moment of the spins. Combining the interactions between spins and the contributions of the external magnetic field, the total energy of a configuration σ is described by the Ising Hamiltonian in Equation (6) [11]. The negative sign in the interaction term ensures that the energy is minimized when neighbouring spins are aligned in the case of ferromagnetic interactions (where J is positive).

3.2 The Ising Problem

The Ising problem involves finding the minimal energy configuration (ground state) of the system described by the Ising model (6). This requires selecting the arrangement of spins σ that minimizes the energy E, which can be used to solve NP-complete and NP-hard problems such as Max-Cut [16] and GCP [15].

3.2.1 Ising Hamiltonian and QUBO Equivalence

The Ising Hamiltonian is a function describing the complete energy landscape that the system's spin configuration seeks to minimize.

The Ising problem can be reformulated as a QUBO problem (and vice versa), where the goal is to minimize a quadratic function of binary variables [16]. This equivalence allows using algorithms and hardware designed for Ising problems QUBO instances.

To transform an Ising problem to a QUBO problem, the spins $\sigma_i \in \{-1, 1\}$ must be mapped to binary variables $x_i \in \{0, 1\}$ using the relation $x_i = \frac{\sigma_i + 1}{2}$ [15]. The Ising Hamiltonian can then be rewritten in terms of binary variables to form a QUBO problem. The reverse transformation is straightforward and requires replacing the binary variables x_i with spin values σ_i using $\sigma_i = 2x_i - 1$.

The Ising problem, known to be NP-complete [45], has high computational complexity due to the exponential number of possible spin configurations (2^N for N spins). This makes it very

computationally expensive to find the ground state for large systems.

3.3 Approaches to Solving the Ising Problem

There are several methods which can be used to solve the Ising problem, each suited to different system sizes and complexities. For smaller systems, exact solvers can be effective, with techniques like brute-force search and ILP [30, 30]. With the increase in problem sizes, these methods become less feasible, and we are forced to use approximation algorithms. These include heuristics and metaheuristics, such as greedy algorithms, SA, genetic algorithms, and neural networks, which provide good quality solutions within a reasonable timeframe, but do not guarantee global optimality.

In addition to classical methods, quantum algorithms offer a different approach. QA and CIMs attempt to find the ground state of the Ising model more efficiently than classical techniques by relying on their hardware's natural tendency to evolve to its minimal energy state [12, 13, 46]. In this work, we focus exclusively on QA.

4 Quantum Annealing

4.1 Overview and Applications

QA is an analogue computing technique used to solve optimization problems by finding the global minimum of an objective function. Unlike gate-based quantum computing, a popular alternative approach used by companies like IBM [47], QA relies on quantum fluctuations and tunnelling to escape local minima and efficiently explore the energy landscape. Quantum fluctuations refer to the temporary changes in energy levels due to the Heisenberg uncertainty principle [48], while quantum tunnelling is a quantum phenomenon allowing particles to pass through energy barriers with classically forbidden regions [49], as shown in Figure 1.

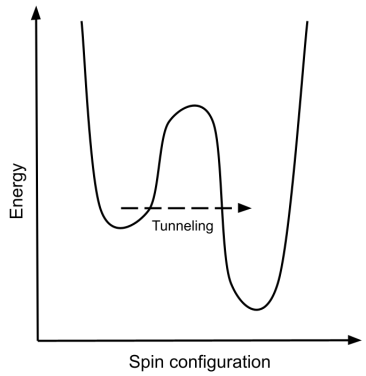

Figure 1: Quantum tunnelling (diagram modified from [50])

The term "annealing" originates from metallurgy, where it describes a process of heating and then slowly cooling a material to remove defects and create a more homogeneous structure [51]. It is also used in various other fields, such as biology and material sciences [52], to describe similar processes of optimization and stabilization through gradual evolution of a system. In quantum systems, annealing involves initializing the system in a simple ground state and slowly evolving it according to the Schrödinger equation towards the ground state of a more complex Hamiltonian, which usually corresponds to the optimal solution of the embedded Ising problem [46, 53].

4.2 Theoretical Principles

In QA, the system is initialized in the ground state of the driver Hamiltonian H_D, which typically corresponds to a simple, easily prepared quantum state [46]. The system's total Hamiltonian during the annealing process is given by:

$$H(t) = A(t)H_D + B(t)H_P \tag{7}$$

where $A(t)$ and $B(t)$ are scheduling functions that control the transition from the driver Hamiltonian H_D to the problem Hamiltonian H_P. Initially at time t_0, $A(t_0) \gg B(t_0)$ holds, making H_D dominant. Over time, $A(t)$ decreases and $B(t)$ increases, so that by the end of the process

at time t_f, $A(t_f) \ll B(t_f)$ holds and H_P governs the system [46].

$$H_P = -J \sum_j \sigma_j^z \sigma_{j+1}^z - \Gamma \sum_j \sigma_j^x \tag{8}$$

Here, J represents the interaction strength between neighbouring qubits, influencing how they align with each other, while Γ controls the tendency of each qubit to explore different states. The Pauli matrices σ^z and σ^x are used because they correspond to measurements in orthogonal bases: the z basis corresponds to the computational basis $\{|0\rangle, |1\rangle\}$, and the x basis corresponds to the orthogonal basis $\{|+\rangle, |-\rangle\}$. The orthogonal bases allow the system to stabilize in computational states (σ^z) and to explore superpositions of states (σ^x), which are required in the quantum tunnelling process driven by H_D. This helps the system to escape local minima and move towards the global minimum solution [46].

By traversing energy barriers that could trap classical algorithms in local minima, quantum tunnelling during the annealing process can lead to more efficient optimization. Given a carefully designed annealing schedule, QA has the potential to find high-quality solutions efficiently, though its computational complexity is problem-dependent [46, 53]. The annealing schedule must balance the need to evolve slowly enough to maintain adiabaticity, ensuring the system stays in its ground state, with the practical need to mitigate decoherence.

On most QA devices, such as those developed by D-Wave, the problem is encoded into the qubit states through the problem Hamiltonian H_P, which can be represented as Equation (7). This H_P is closely related to the Ising model (6), which describes the interactions between neighbouring spins in the system.

The system's evolution is guided by the combination of the driver Hamiltonian and the problem Hamiltonian [40]. Entanglement between qubits is a natural consequence of the interactions defined by the problem Hamiltonian and is necessary for achieving quantum advantage in some cases [40, 41]. Finally, the qubits are measured in the computational basis to determine the final state after annealing.

As previously mentioned, one of the key challenges in QA is mitigating decoherence by ensuring maximal system isolation. Coupling with the environment can introduce unwanted interactions that disrupt the annealing process and reduce the likelihood of the system reaching the ground state. To counteract these effects, QA systems are typically shielded and cooled to a few millikelvin [40], minimizing interactions with thermal photons, electromagnetic interference, and other sources of decoherence [54, 55]. It is important to note that the ground state in QA is particularly susceptible to decoherence caused by thermal excitations, which can depopulate the ground state and negatively impact the annealing process [54].

4.3 D-Wave Quantum Annealer

4.3.1 Current Devices

So far, D-Wave has had two commercially available Quantum Processing Unit (QPU) systems: the Q-2000 and the D-Wave Advantage. Introduced in 2017, the Q-2000 features over 2,000 qubits and more than 6,000 couplers, using a Chimera qubit graph topology that allows for a maximum 6-way qubit connectivity [56, 27, 37]. Their latest QPU, the D-Wave Advantage, was launched in 2022 and boasts significant hardware improvements, including over 5,000 qubits, 35,000+ couplers, and improved 15-way connectivity due to the Pegasus qubit graph topology [42, 37, 27]. Both systems are accessible through D-Wave's cloud computing services and use superconducting materials for their qubits [55].

On D-Wave devices, entangled qubits are created using superconducting flux qubits that operate at temperatures below 20 mK [55], with interactions controlled by magnetic flux [40]. The QPU's couplers enable these interactions, and the connectivity metric indicates how many other qubits each qubit can interact with. Unlike in gate-based quantum computing, these interactions are unique to the problem Hamiltonian and cannot be turned off, similarly to condensed-matter systems where entanglement naturally arises from the system's configuration during the annealing process [40].

4.3.2 Software

To facilitate interaction with their QPUs, D-Wave provides the open-source Ocean software suite, which includes tools for problem formulation, embedding, and execution. The suite also includes a range of solvers that can run on local hardware, along with HA solvers that combine classical and quantum algorithms to solve CO problems. Notably, these hybrid solvers are executed entirely on D-Wave's compute infrastructure. In this work, we use D-Wave's Ocean software suite to benchmark and compare their SA, HA and QA solvers.

4.3.3 Quantum Advantage

There have been several claims of quantum advantage demonstrated by D-Wave systems, such as the scaling advantage against Monte-Carlo Path integrals [57]. There, D-Wave's quantum devices are reported to have shown a significant scaling advantage in simulating geometrically frustrated magnets, with a reported million-fold speed-up compared to classical path-integral Monte Carlo methods, particularly as system size and inverse temperature increase [57].

Additionally, there are claims that D-Wave's quantum annealer exhibits a scaling advantage over classical SA, showing better performance on problem sizes up to 2000 qubits, though SA still appears to have the best overall scaling [58].

Comparisons have also been made indicating that D-Wave's quantum annealer achieves higher success probabilities and lower energy levels than IBM's digitized quantum annealing, challenging previous claims of IBM's superior performance [59].

5 Optimizing Wavelength Allocation (WA)

WA is the process of assigning specific wavelengths to routes in WDM optical networks, ensuring that no two lightpaths that share the same fibre are assigned the same wavelength [4]. The optimization goal of WA is to minimize the number of wavelengths used [5, 10]. In this context, we focus on WS optical networks utilizing FXC and WSXC cross-connects [6], where each lightpath remains on a single wavelength throughout its route.

Optical multiplexers and demultiplexers are devices that enable simultaneous multi-wavelength communication through a single fibre cable. A multiplexer combines several independent signals at different wavelengths onto a single optical fibre, while a demultiplexer separates them back into individual source signals at the destination. These devices operate using wavelength-selective components, such as diffraction gratings or filters, to isolate or combine specific wavelengths [60].

Minimizing the total number of wavelengths is desirable for reducing the cost and complexity of network components, since devices capable of supporting a larger number of wavelengths are typically more difficult and expensive to manufacture. Additionally, using fewer wavelengths allows for easier network expansion and accommodating increased lightpath demand [9, 5].

The lower and upper bounds on the number of wavelengths depend on factors such as the network's topology, the requested number of lightpaths, and the routing configurations. The lower bound represents the theoretical minimum number of wavelengths required, while the upper bound reflects the worst-case scenario of maximum wavelength usage. In densely connected networks, where many lightpaths share the same links, more wavelengths are needed to avoid interference.

WA can be classified into static and dynamic approaches [10]. In static WA, wavelengths are pre-allocated according to a predefined routing plan, ensuring that each wavelength is reserved for specific routes prior to any data transmission. This method is ideal for networks where traffic patterns are predictable and stable over time. On the other hand, dynamic WA adapts to real-time changes in network conditions, such as fluctuating traffic loads or link failures, making it more suitable for rapidly changing environments like core or metropolitan networks [9].

Current industry solutions for WA focus on heuristic and exact methods. Heuristic approaches, like the greedy LDF algorithm [7], offer quick, not-always-optimal solutions, while exact methods, such as ILP solvers, provide optimal results but may not scale well for large networks [30]. However, traditional approaches often struggle with the increasing complexity of modern optical networks, leading to challenges in scalability, runtime, and memory requirements [28].

Recent research has explored novel methods for solving the WA problem, particularly through quantum and quantum-inspired [2] approaches. For instance, simulated CIMs [18] and D-Wave's quantum annealers [27] can be used to solve WA, as the problem can be expressed in QUBO form and embedded on a QPU [2, 19].

5.1 QUBO Formulation

The WA problem in a WDM optical network can be mapped to the GCP. In this mapping, each route corresponds to a node, and edges are drawn between nodes representing routes that share a common fibre cable [2]. The requirement that no two routes on the same link share a wavelength translates to the GCP constraint that no two adjacent nodes share the same colour. The goal of minimizing the number of wavelengths corresponds to finding the chromatic number of the graph, which is the smallest number of colours needed to colour all the nodes. Figure 2

(extracted from [2]) illustrates the transformation of the WA problem in a simple linear network topology with 5 nodes and 4 links into GCP form, and the subsequent conversion back to WA after solving the GCP.

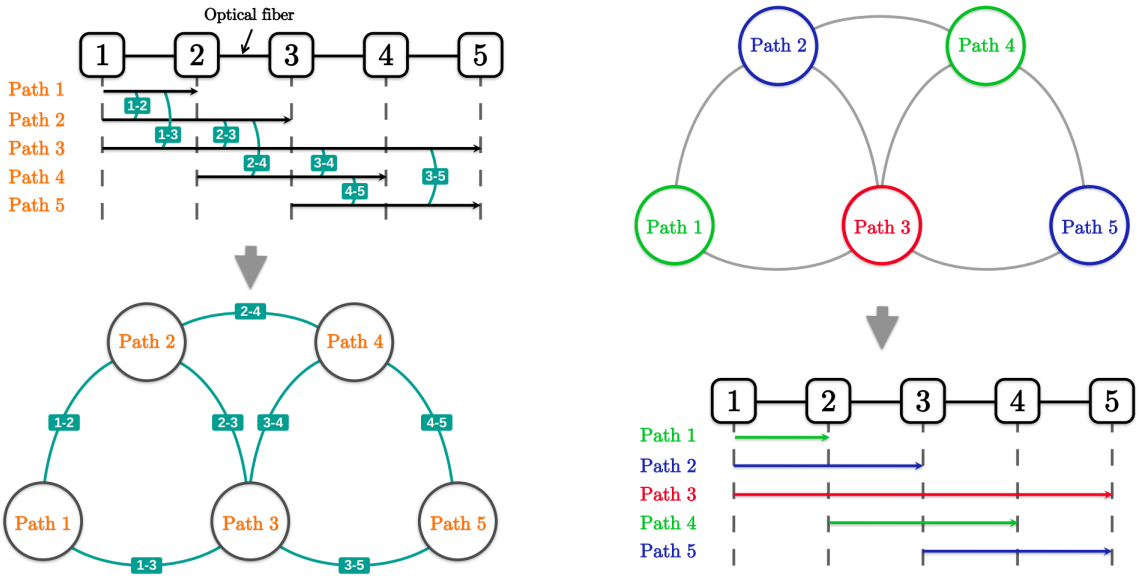

Figure 2: WA problem transformation to GCP (image extracted from [2])

To transform the GCP into a QUBO formulation, [15, 2] express the problem's constraints using binary variables:

$$x_{ij} = \begin{cases} 1 & \text{if node } i \text{ is assigned colour } j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$k_i = \begin{cases} 1 & \text{if colour } i \text{ is assigned} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The requirement that only one colour is assigned per node is expressed as [15]:

$$\sum_{j=1}^{K} x_{ij} = 1,\ i = 1, ..., n \tag{11}$$

where n is the number of nodes in the graph and K is the number of colours. To transform this constraint into QUBO form, we apply Transformation 1, as described in [15]. This converts equality constraints of the form $Ax = b$ into a QUBO model by introducing a quadratic penalty term $P(Ax-b)^t(Ax-b)$. Using this transformation, we derive the following quadratic expression:

$$Q_1 = P_1 \left(2 \sum_{1 \le j < l \le K} x_{ij} x_{il} - \sum_{j=1}^{K} x_{ij} \right) \tag{12}$$

Here, P_1 is a positive penalty parameter that enforces the constraint by penalizing solutions

violating the requirement of Equation (11) [15].

The constraint that all pairs of neighbouring nodes i, j must have different colours is encoded as such [15]:

$$x_{ip} + x_{jp} \leq 1,\ p = 1..., K \tag{13}$$

This constraint is formulated into QUBO form using Transformation 2, as described in [15]. This transformation is a special case of Transformation 1 which handles inequality constraints of the form $x_i + x_j \leq 1$ by converting them into a penalty term $P(x_i x_j)$, resulting in the following term:

$$Q_2 = P_2 \sum_{(i,j)\in E} \sum_{p=1}^{K} x_{ip} x_{jp} \tag{14}$$

In this case, P_2 is also a positive penalty parameter that penalizes assignments of the same colour to adjacent nodes, enforcing the graph colouring constraint described in equation (13).

Finally, the combined QUBO objective function is given by minimizing the sum of these terms [15]:

$$\min_x x^t Q x = \min_x (Q_1 + Q_2) \tag{15}$$

The minimization criteria for finding a configuration with the least number of colours is expressed as such [2]:

$$\sum_{i=1}^{K} k_i \to \min \tag{16}$$

An Ising Hamiltonian for the GCP given all adjacent nodes i, j, can be constructed as such [2]:

$$H(x) = H_1(x) + H_2(x) \tag{17}$$

$$H_1(x) = \sum_{v=1}^{n} (1 - \sum_{u=1}^{K} x_{vu}) 2 \tag{18}$$

$$H_2(x) = \sum_{i,j} \sum_{p=1}^{K} x_{ip} x_{jp} \tag{19}$$

5.2 Results and Analysis

We conducted benchmarks for various CO solvers on the WA problem, using several real-world network topologies. The topologies tested include Eurocore, EON, ARPANET (1972), UKnet, Eurolarge, and USnet (see Appendix 8.3 for network plots). Routing configurations, defined as subsets of the set of all shortest routes between all node pairs, were randomly selected from a uniform distribution of all possible routes across six different degrees of the route density. Here, "route density" refers to the proportion of the total possible routes selected in the configuration and is referenced in Figure 3. For each benchmark, routes are incrementally added to the subset as density increases. This means that for a given route density (e.g., 0.2, meaning 20% of the total possible routes), the selected routes remain fixed and are incrementally expanded as density increases. This approach allows us to evaluate how different solvers perform as network WA increases.

For the D-Wave solvers, the QUBO problem was first transformed into an Ising Hamiltonian seen in Equation (17), which was then embedded onto a D-Wave QPU or simulator using the Ocean SDK. For the Gurobi solver, which finds globally optimal solutions using an linear

programming-based branch-and-bound method [61], we use an existing ILP formulation of the GCP problem (see source code for details, link in Appendix 8.2). The greedy LDF algorithm was directly applied to the target graph using the NetworkX Python package.

Solvers requiring local computation were executed with a maximum runtime of 60 minutes on an AMD Ryzen 9 5900HX CPU. The D-Wave Leap Hybrid Annealer and Quantum Annealers were accessed through D-Wave's cloud compute platform, with each job allocated a response timeframe of 60 minutes.

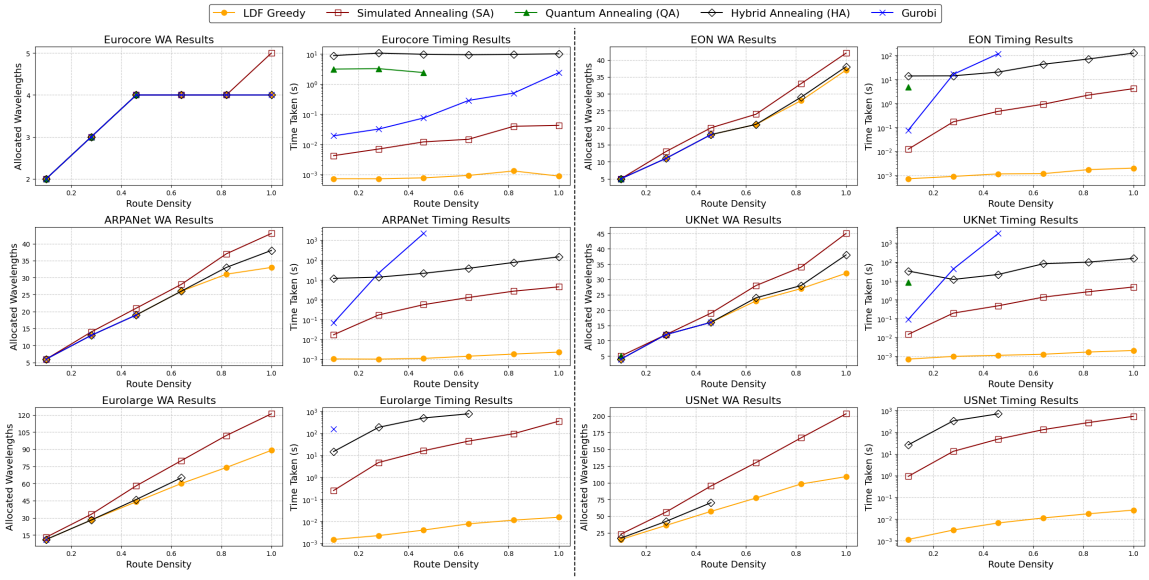

Figure 3: WA problem: Total wavelengths assigned and time taken by different solvers

Figure 3 compares the performance of various solvers across different network topologies, highlighting both solution quality (measured by the number of wavelengths assigned) and computation time. The solvers tested include Simulated Annealing (SA), Quantum Annealing (QA), and Hybrid Annealing (HA)—all developed by D-Wave, ranging from the fully quantum QA to the quantum-classical HA, and the quantum-inspired SA. For comparison, the fully classical greedy LDF heuristic and the exact Gurobi solver are also included. The figure is organized into six groups, each containing two plots: the left plot shows the total number of wavelengths assigned for a given network configuration, while the right plot provides a logarithmic representation of the computation time. Both plots contain data from the same benchmark run. The x-axis of each plot represents route density, defined as the proportion of total possible routes selected in the configuration. Each group corresponds to a different network topology, arranged by node count, from the smallest (Eurocore) in the top-left to the largest (USNet) in the bottom-right. If a curve is missing or terminates before reaching the end of the x-axis, it indicates that the solver either timed out without producing a solution or the problem was too large to be embedded on the QPU. Table 1 presents information about the number of nodes and edges in the transformed graphs of the various networks at different levels of route densities.

Network Name	Route Density	Nodes	Edges
Eurocore	10%	11	2
	28%	31	15
	46%	51	53
	64%	70	99
	82%	90	158
	100%	110	224
EON	10%	38	52
	28%	106	466
	46%	175	1331
	64%	243	2377
	82%	312	3903
	100%	380	5834
ARPANET (1972)	10%	38	60
	28%	106	568
	46%	175	1645
	64%	243	3007
	82%	312	5037
	100%	380	7664
UKNet	10%	42	52
	28%	118	583
	46%	193	1555
	64%	269	3139
	82%	344	5311
	100%	420	7829
Eurolarge	10%	181	880
	28%	506	7333
	46%	831	19810
	64%	1156	39533
	82%	1481	65272
	100%	1806	97930
USNet	10%	207	1999
	28%	580	15970
	46%	952	41524
	64%	1325	81710
	82%	1697	132004
	100%	2070	193890

Table 1: WA transformed graph characteristics

5.3 Discussion

The benchmark results indicate that the Gurobi solver and the greedy LDF heuristic consistently allocate the fewest wavelengths, thus producing the highest quality solutions. However, Gurobi struggles to generate solutions within the allocated 1-hour time frame when the transformed graph exceeds 200 nodes and 1,500 edges. In the benchmark presented Gurobi and greedy LDF provide the same globally optimal solution in all the cases. Nevertheless, when testing previous runs cases where the LDF heuristic produced non-globally optimal results were observed.

The HA solver typically assigns on average 7.1% more wavelengths per solution than Gurobi and greedy LDF, and is capable of solving larger graphs with sizes up to 1,000 nodes and 40,000 edges. QA performs similarly to HA, allocating 5.6% more wavelengths than Gurobi, but is significantly more constrained by the size of the problems it can handle. For example, QA fails to produce solutions for even the smallest network, Eurocore, when the route density exceeds 46% (70 nodes, 99 edges), primarily due to the physical limitations of the QPU's embedding size. In contrast, HA uses classical algorithms to reduce the embedding size for the QPU, allowing it to solve larger problems. Lastly, SA consistently produces the lowest quality solutions, requiring 41.7% more wavelengths than the LDF heuristic. However, the only solvers that reliably solve the largest problems in the dataset are SA and greedy LDF.

Regarding computation time, the greedy LDF heuristic consistently performs best, with times in the millisecond range for all benchmarked instances, including an impressive 26 milliseconds for the 2,070 node and 193,890 edge USNet configuration. SA follows, taking 42 milliseconds for the highest route density in the Eurocore network and 9 minutes for the full-density USNet. Gurobi outperforms QA and HA for graphs with fewer than 100 nodes and 500 edges, but its computation time increases exponentially, making it unable to produce solutions for graphs with 200 nodes and 1,500 edges within the 1-hour limit. For the few cases where it does generate solutions, QA outperforms HA by a factor of 3. In terms of raw QPU compute time, QA manages to solve GCP for a 50 node and 53 edge graph in 53 milliseconds. HA's time complexity follows a similar path to that of SA; however, for graphs with more than 1,000 nodes and 40,000 edges, it also fails to produce solutions within the allotted time. Moreover, the quantum-inspired SA outperforms HA in computation time by an average factor of 340, although this advantage is likely affected by network delays and queue times on D-Wave's cloud compute infrastructure.

6 Optimizing Routing and Wavelength Allocation (RWA)

Routing is a fundamental problem in graph theory with numerous practical applications, particularly in computer networks, logistics, and telecommunications. In the context of computer networks, the routing problem involves determining the most efficient paths for data transmission between nodes [8]. Common algorithms used to solve routing problems on weighted or unweighted graphs include Dijkstra's algorithm and different heuristic or metaheuristic approaches like Genetic Algorithms for more complex scenarios [62].

RWA is a combined problem that involves allocating efficient paths for data transmission while also assigning wavelengths to in a way which minimizes their total number [5]. While routing typically prioritizes metrics like shortest path or highest bandwidth, RWA may select a less direct route to minimize congestion on individual links, thus reducing the number of wavelengths required. In the QUBO formulation presented later in this section, secondary criteria such as bandwidth capacity can be incorporated.

An important real-world problem in RWA is handling link failures. Static routing precomputes path configurations and lacks real-time adaptability, making it less flexible unless failure scenarios have been precomputed as well. In contrast, dynamic routing can adjust to changes like link failures and variations in lightpath demand by recalculating optimal routes as needed, offering greater resilience [9]. However, this adaptability often comes at the cost of increased computational complexity.

Like WA, solving RWA can be achieved using exact algorithms such as ILP, greedy heuristic algorithms, and more recently, quantum techniques [1]. Industry solutions often rely on combinatorial solvers like Gurobi or heuristics tailored to specific topologies, which face the same limitations discussed in the WA section.

QC approaches offer a new way for solving RWA problems, with the potential for achieving shorter computation times in the future compared to classical methods. One study demonstrated the feasibility of using QUBO formulations to solve dynamic RWA and other optical routing problems with a hybrid quantum annealer, noting that repeated sampling was often required to achieve optimal solutions due to the heuristic nature of the solver [1]. The formulation presented in this work uses QA, HA, and SA to solve static RWA, specifically targeting the optimization of the routing stage to minimize the number of wavelengths needed in the subsequent WA stage. Our method introduces a novel approach to solving the static RWA problem using precompiled optimal routes. The QUBO formulation developed is applicable to multiple solvers, including SA, HA, and QA, and we benchmark the results across them all.

6.1 QUBO Formulation for RWA

In this section, we present a QUBO formulation for solving the RWA problem in optical networks. We begin with a network topology and a set of (source, destination) node pairs, for which we aim to determine the optimal routes and corresponding wavelength assignments. This set can include all possible (source, destination) pairs if we wish to solve for all routes in the network, but in realistic scenarios we would likely only need the subset of the most commonly used routes.

For each (source, destination) pair, we precompute the top N_b routes based on an optimization criteria such as the lowest hop counts. These routes are then represented as nodes in a new graph, where an edge between any two nodes indicates that the corresponding routes share one or more links in the original network. The goal is to select a configuration of routes where we have exactly one route for each source-destination pair, and the selected configuration minimizes the

sum of shared links between routes, represented by the edges in the transformed graph. When we have a configuration, we calculate the total number of edges in the selected subgraph, which corresponds to the amount of shared links between the routes from the configuration. Since we aim to minimize this value, we must encode it in the matrix Q of the QUBO formulation.

By minimizing the shared links, we reduce congestion in the network, which in turn simplifies the WA problem, leading to more efficient utilization of network resources.

6.2 Initial Graph Transformation

The graph transformation of the initial network topology is based on the set of routes for which we want to solve the RWA problem, corresponding to specific (source, destination) pairs. For each of these pairs we have precomputed an N_b number of "best routes", representing the most optimal paths from the source to the destination based on criteria such as hop count. When creating the transformed graph, each of the N_b precomputed best routes then becomes a node, which is why we use nodes and routes interchangeably when referring to solving the QUBO problem on the transformed graph. If two routes share a link in the initial network at any point, this relationship is represented as an edge between their respective nodes in the transformed graph. This transformation allows us to solve the RWA problem by converting it into a graph problem, where our goal is to select a subset of nodes which minimizes the sum of their node degrees with other nodes in the solution set. This process reduces the total number of shared links among the selected routes in the original network, resulting in a subgraph of selected nodes. This subgraph is already in the GCP graph form (as shown in Figure 2), allowing us to solve the WA directly. However, as shown in Figure 5, even small examples can transform into highly connected graphs that grow exponentially with the size of the initial network.

In our approach, the routing problem is partially solved by precomputing the N_b best routes between each source-destination pair. Since a trivial solution to the problem is to output an empty set of node selections (thus having a total of 0 node degrees), the key challenge becomes how to adjust penalty coefficients in order to select exactly one route for each source-destination pair from these precomputed routes. This is done by encoding the constraint that exactly one route must be selected per source-destination pair in the QUBO formulation by applying a heavy penalty for choosing zero or more than one routes. We also encode the nodal degrees in the QUBO problem such that minimizing $s^T Q s$ will naturally tend towards a less-congested routing configuration. Due to the unconstrained nature of QUBO, the solution's validity must still be verified after the optimization run.

Figure 4 illustrates a simple network topology, Table 2 lists the information about the different precomputed best routes of the graph and Figure 5 depicts the transformed graph where the best routes are represented as nodes and shared links as edges.

Route ID	Src ID	Dest ID	Path (Link IDs)
0	1	2	1
1	1	2	2-3
2	1	3	2
3	1	3	1-3
4	1	4	2-4
5	1	4	1-3-4
6	2	1	1
7	2	1	3-2
8	2	3	3
9	2	3	1-2
10	2	4	3-4
11	2	4	1-2-4
12	3	1	2
13	3	1	3-1
14	3	2	3
15	3	2	2-1
16	3	4	4
17	4	1	4-2
18	4	1	4-3-1
19	4	2	4-3
20	4	2	4-2-1
21	4	3	4

Table 2: Routing Table

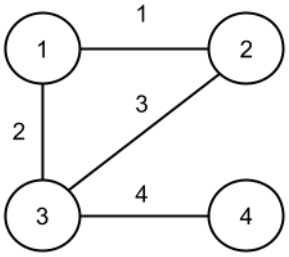

Figure 4: Simple network topology

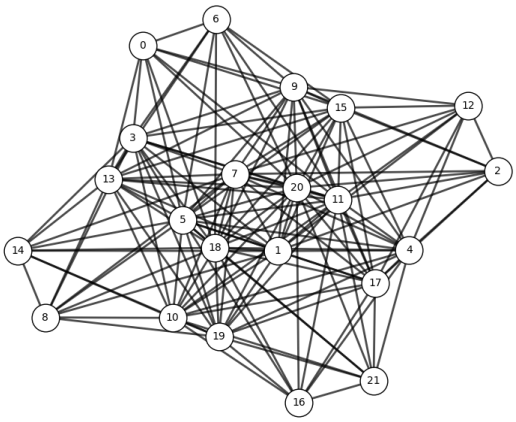

Figure 5: Transformed graph

6.3 Mathematical Formulation

In this section, our novel mathematical formulation of static RWA is described. We discuss "best routes" in the context of routes containing the fewest hop counts, but generally, a different quantifiable route metric can be used if needed.

Let N_n be the number of nodes in the original network.
Let N_b be the number of precomputed best routes for each valid (source, destination) node pair.
Let P be the set of all source, destination node pairs in the original network.
Let $B = \{1, ..., N_b\}$, where N_b is the number of precomputed best routes.
Let R be the set of all best routes, $|R| = |P| \times N_b$.
Let G be the transformed graph.

Defining function $f : P \times B \xrightarrow{f} R$, which maps each of the top N_b best routes for a given (source, destination) pair to a unique route in the set R.
Defining function $h : R \to \mathbb{Z}^+$, which gives a route's hop count.

$$\forall (i,j) \in P \times B, \exists! r \in R \; s.t. \; f(i,j) = r \tag{20}$$

$$a_{nm} = \begin{cases} 1 & \text{if } (n,m) \in E \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

$$x_i = \begin{cases} 1 & \text{if route i is used} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

$$h_i = \sum_{k=0}^{M-1} 2^k b_{i,k}, \quad b_{i,k} \in \{0,1\} \tag{23}$$

$$H_0 = \sum_{i \in R} h_i = \sum_{i \in R} \sum_{k=0}^{M-1} 2^k b_{i,k} \tag{24}$$

$$H_1 = \sum_{i \in P} \left(1 - \sum_{j \in B} x_{f(i,j)}\right)^2 = \sum_{i \in P} \left(\sum_{j,k \in B} x_{f(i,j)} x_{f(i,k)} - 2 \sum_{l \in B} x_{f(i,l)} + N_p\right) \tag{25}$$

$$H_2 = \sum_{n \in P} \sum_{m \in P} a_{nm} x_n x_m \tag{26}$$

$$H = c_0 H_0 + c_1 H_1 + c_2 H_2 \tag{27}$$

We encode the condition that for every best route for valid pairs, $(i,j) \in P, i \neq j$ there exists a unique route $r \in R$ as shown in Equation (20). The adjacency of nodes n, m in the transformed graph G is given by the binary variable a_{nm}, defined in Equation (21). The binary variable x_i indicates whether a route is selected, as defined in Equation (22).

Next, the Hamiltonians for minimizing the route hop count, encoding the condition of selecting only one route per source-destination node pair, and minimizing node degrees are given by Equations (24), (25), and (26), respectively. To map this problem to a QUBO form, we first expand H_1 as shown in Equation (25), and encode the hop count function h_i in binary form as shown in Equation (23). We then reformulate H_0 using binary variables, as demonstrated in Equation (24). Since this component of the Hamiltonian is designed to minimize secondary routing criteria (in this case, hop count), adjusting the parameter c_0 in Equation (27) allows us to choose a balance between an optimal WA solution and one that reduces hop counts. Similarly, the parameters c_1 and c_2 control the one route per (source, destination) pair constraint and the nodal degree minimization objective respectively. In a more general case, hop counts can be substituted with other weights associated with individual links, such as bandwidth capacity, allowing the embedding of more complex secondary criteria.

Once fully formulated using binary variables, the Hamiltonian transforms into a QUBO problem that can be represented in quadratic form as $s^T Q s$, where s is a vector of binary variables and Q is a matrix of quadratic coefficients. As with all QUBO problems, the goal is to minimize $s^T Q s$ by finding the optimal binary vector s. We construct the matrix Q for routes $i, j \in R$ as follows:

$$Q_{ij} = \begin{cases} c_0 H_0 & \text{if } i = j \\ c_1 H_1 & \text{if } i \neq j \text{ and } i,j \text{ are routes for the same source-destination pair} \\ c_2 H_2 & \text{if } i \neq j \text{ and } i,j \text{ share a link} \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

The implementation details regarding the construction of this matrix can be found in the project source code (see Appendix 8.2) located at `src/connected_network.py`, in the `ConnectedNetwork.solve_qubo` method.

6.4 Results and Analysis

We solve the static RWA on a toy problem topology with 6 nodes and 20 edges (reverse links are distinct), shown in Figure 6. The transformation which encodes the shared fibre relationships between the precomputed routes is shown in Figure 7. In this example, we consider reverse links to be distinct, i.e., route $1 \rightarrow 2$ is different from route $2 \rightarrow 1$. Thus, we calculate the total nodes in the transformed graph: $N_n(N_n - 1)N_b = 6 \cdot 5 \cdot 3 = 90$, where N_n is the number of nodes in the initial network, and N_b is the number of lowest hop count routes considered between each node pair. The resultant graph contains 90 nodes and 694 edges, which shows very large problem size increase even for a small initial topology.

We run the optimization algorithm for the resultant QUBO problem on three D-Wave samplers: SA, QA, and HA. All three samplers produced well-optimized solutions in terms of minimizing nodal degrees and maximizing low-hop route selection. The selected routes are then made into a graph for which the GCP is solved using QA, producing the optimal result of a total 3 wavelengths allocated, seen in Figure 8 (route and solution tables available in Appendix 8.4). The optimal result was verified using a simple brute-force search, which ensures global optimality. We also calculate the probability of selecting a route depending on its hop count for all three approaches, averaging the results over 100 iterations. Results show very similar route selection probabilities for all three solvers, displayed in Figure 9.

In terms of runtime performance, the QA solver averaged approximately 48 ms of direct QPU computation over 20 runs for the toy problem, given well-optimized parameters. The processing time for parameter optimization using grid search on the toy problem varied depending on the grid parameter spacing, but finding reliable, high-quality solutions typically required 15 minutes of computation.

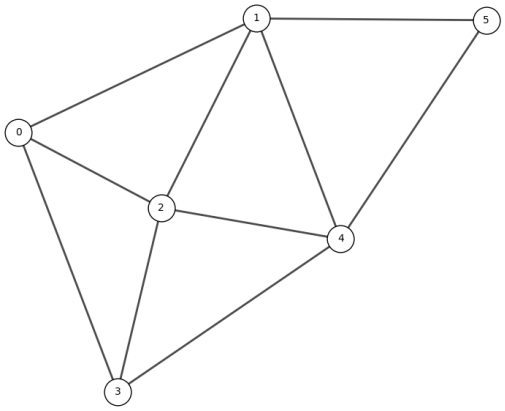

Figure 6: Toy problem topology

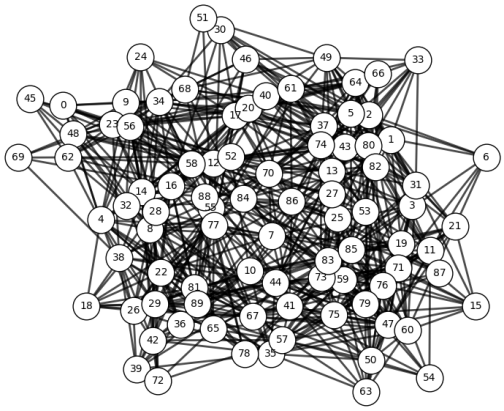

Figure 7: Transformed toy problem graph

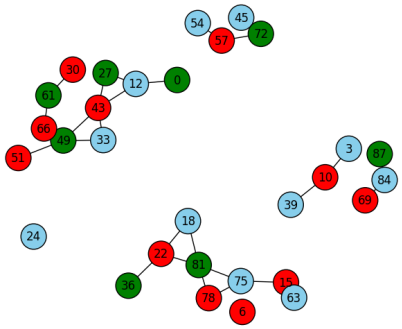

Figure 8: RWA solution

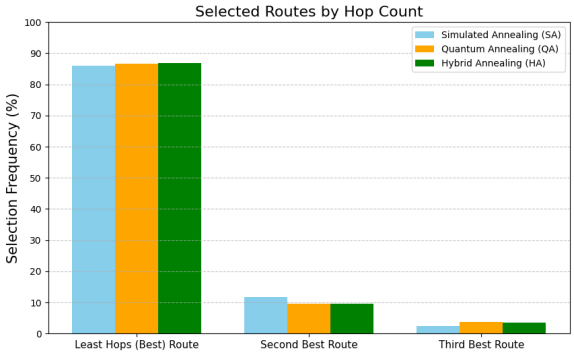

Figure 9: Route selection histogram

6.5 Discussion

This formulation aims to provide a way to find routing configurations that evenly distribute link utilization across the network, reducing congestion and the overall allocated wavelength count. However, formulating the problem as a QUBO results in exponential growth in problem size, directly correlated with the network's scale. Even relatively simple networks, such as ARPANET (1970) with only 20 nodes, result in a graph transformation with over 1,500 highly connected vertices. Most quantum and quantum-inspired solvers are unable to reliably solve problems of this size within a reasonable timeframe.

To get valid solutions reliably, parameter optimization of c_0, c_1, c_2 from Equations (27) and (28) was necessary, performed via grid search and followed by evaluation of SA outcomes to determine overall solution quality. SA was chosen for parameter optimization as opposed to HA and QA, since it performs local computations only and does not need to make requests to D-Wave's cloud compute. This significantly reduces the time required for SA to find the optimal configuration. Once optimized, both HA and QA were able to solve the QUBO problem, producing solutions with similar distributions in regard to the secondary optimization criteria. Nevertheless, similarly to previous work, repeated sampling was required to reliably obtain valid solutions, with fewer samplings needed when better optimized parameters were used [1].

Improvements to this formulation could be achieved by reducing the number of dynamical variables in the Hamiltonian and applying classical heuristics to preprocess the problem graphs and eliminate inefficient routes (e.g., excessive hops or shared links). Improving the graph transformation and Hamiltonian minimization criteria could also be beneficial. Further optimizing the parameters c_0, c_1, c_2 can lead to more reliably reproducable solutions, as the solution quality is highly sensitive to these values. However, this optimization process can quickly become computationally expensive, particularly for larger problems. As mentioned, the toy problem parameter optimization process took upwards of 15 minutes for finding good quality parameters.

7 Conclusion and Future Work

7.1 Conclusion

This work has reviewed and benchmarked the performance of SA, HA, and QA in solving WA and RWA problems, particularly within the framework of static problems using wavelength division multiplexing. Through the introduction of a novel QUBO formulation for the RWA problem, the research analyses challenges caused by the exponential growth in problem size as network scale increases.

The benchmarking results show that while the Gurobi solver and the greedy LDF heuristic solve WA with the smallest number of allocated wavelengths, Gurobi's performance deteriorates with larger problem sizes, likely due to the complexity of the problem constraints. Although the HA solver is capable of handling comparably sized problems, its solution quality is lower than those produced by classical solvers. The QA solver is even more restricted by problem size due to physical limitations of the QPU. In all timing benchmarks, QA outperforms HA in terms of processing time, while SA always outperforms both HA and QA (Figure 3) by a factor of 300. SA consistently produces the lowest quality solutions out of all the solvers. However, SA and greedy LDF show to be the only solvers capable of consistently solving large-scale problems. Overall the LDF heuristic provides the best balance between solution quality and computation time.

For the static RWA problem, the study demonstrates that the QUBO formulation produces routing configurations that distribute link utilization evenly across the network. However, the formulation exhibits exponential growth of the transformed problem graph, which limits the capability of quantum and quantum-inspired solvers to reliably solve the QUBO problems. Parameter optimization is necessary to reliably generate valid solutions, but it becomes a significant challenge in networks of more than 10 nodes due to its high computational complexity.

Despite these challenges, the proposed QUBO formulation provides a novel approach for optimizing network configurations for RWA problems and may become computationally feasible for application to real-world problems if the scalability issues are addressed.

7.2 Future Work

There are several potential improvements that could be made to the current RWA QUBO formulation. One key area is improving the problem size scaling to make it more manageable for solving larger networks. Machine learning techniques could also be explored to optimize the parameters of the RWA formulation more efficiently, as similar methods have been successfully applied to QUBO problems [63]. Additionally, using heuristic methods for problem preprocessing could help prune inefficient solution trees and improve computational efficiency. Finally, investigating quantum-based approaches for solving RWA with protection could enable fault-tolerant routing and improve network resilience against failures.

7.3 Dataset Information

In this work, we use several real-world network topologies to benchmark solutions for WA and RWA problems. The networks include Eurocore, EON, ARPANET (1970), UKnet, Eurolarge, and USnet (see Appendix 8.3 for more information). The network topology files and their precomputed optimal routes are available in this project's GitHub repository `https://github.com/r1ghtwr0ng/rwa_qubo`.

7.4 Source Code Availability

The source code developed and used in this work is available at the following repository: `https://github.com/r1ghtwr0ng/rwa_qubo`. The code is licensed under the MIT Licence. Instructions for setting up the dev environment and running the code are provided in the repository's `README.md` file.

8 Appendix

8.1 Abbreviations

Acronyms

CIM	Coherent Ising Machine
CO	Combinatorial Optimization
FXC	Fibre Switch Cross-Connect
GCP	Graph Colouring Problem
HA	Hybrid Annealing
ILP	Integer Linear Programming
LDF	Largest Degree First
MILP	Multiple-Integer Linear Programming
NISQ	Noisy Intermediate-Scale Quantum
RWA	Routing and Wavelength Allocation
SA	Simulated Annealing
QA	Quantum Annealing
QC	Quantum Computing
QPU	Quantum Processing Unit
QUBO	Quadratic Unconstrained Binary Optimization
WA	Wavelength Allocation
WDM	Wavelength Division Multiplexing
WS	Wavelength-Selective
WSXC	Wavelength-Selective Cross-Connects

8.2 Source Code

The source code used in this work is publicly available under the MIT Licence at:
`https://github.com/r1ghtwr0ng/rwa_qubo`

8.3 Network Topologies

Please note that the network topology plots in Figure 10 are not accurate in regard to the geographical placement of the real-life network nodes. The topology files are included in the project repository. Table 1 provides information about the number of nodes and edges in the different network topologies.

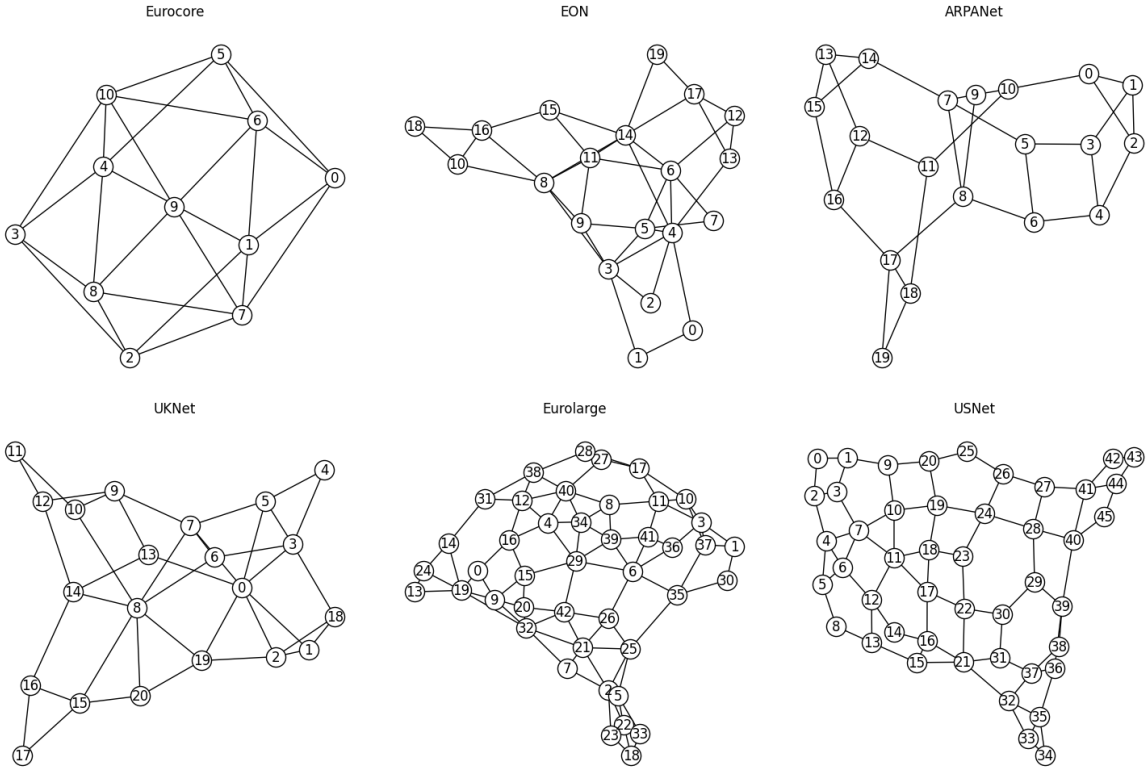

Figure 10: Network topology plots

Network Name	Nodes	Edges
Eurocore	10	15
EON	20	25
ARPANET (1972)	30	35
UKNet	40	45
Eurolarge	50	55
USNet	60	65

Table 3: Summary of network topologies nodes and edges

8.4 QUBO static RWA results

Route ID	Src ID	Dest ID	Path (Link IDs)
0	0	1	1
1	0	1	2-9
2	0	1	3-13-9
3	0	2	2
4	0	2	1-5
5	0	2	3-13
6	0	3	3
7	0	3	2-10
8	0	3	1-5-10
9	0	4	1-6
10	0	4	2-11
11	0	4	3-14
12	0	5	1-7
13	0	5	2-9-7
14	0	5	1-6-18
15	1	0	4
16	1	0	5-8
17	1	0	6-16-8
18	1	2	5
19	1	2	4-2
20	1	2	6-16
21	1	3	4-3
22	1	3	5-10
23	1	3	6-17
24	1	4	6
25	1	4	7-20
26	1	4	5-11
27	1	5	7
28	1	5	6-18
29	1	5	5-11-18
30	2	0	8
31	2	0	9-4
32	2	0	10-12
33	2	1	9
34	2	1	8-1
35	2	1	11-15
36	2	3	10
37	2	3	8-3
38	2	3	11-17
39	2	4	11
40	2	4	9-6
41	2	4	10-14
42	2	5	11-18
43	2	5	9-7
44	2	5	11-15-7

Route ID	Src ID	Dest ID	Path (Link IDs)
45	3	0	12
46	3	0	13-8
47	3	0	14-15-4
48	3	1	12-1
49	3	1	13-9
50	3	1	14-15
51	3	2	13
52	3	2	12-2
53	3	2	14-16
54	3	4	14
55	3	4	13-11
56	3	4	12-1-6
57	3	5	14-18
58	3	5	12-1-7
59	3	5	14-15-7
60	4	0	15-4
61	4	0	16-8
62	4	0	17-12
63	4	1	15
64	4	1	16-9
65	4	1	18-19
66	4	2	16
67	4	2	15-5
68	4	2	17-13
69	4	3	17
70	4	3	16-10
71	4	3	15-4-3
72	4	5	18
73	4	5	15-7
74	4	5	16-9-7
75	5	0	19-4
76	5	0	20-15-4
77	5	0	19-5-8
78	5	1	19
79	5	1	20-15
80	5	1	20-16-9
81	5	2	19-5
82	5	2	20-16
83	5	2	19-4-2
84	5	3	20-17
85	5	3	19-4-3
86	5	3	20-16-10
87	5	4	20
88	5	4	19-6
89	5	4	19-5-11

Table 4: Toy problem routing table

Table 5 contains the route index from Table 4 allocated to the (source, destination) pair in the solution. It also provides the ID of the wavelength allocated for the lightpath.

Source ID	**Destination ID**	**Route ID**	**Wavelength ID**
0	1	0	0
0	2	3	1
0	3	6	2
0	4	10	2
0	5	12	1
1	0	15	2
1	2	18	1
1	3	22	2
1	4	24	1
1	5	27	0
2	0	30	2
2	1	33	1
2	3	36	0
2	4	39	1
2	5	43	2
3	0	45	1
3	1	49	0
3	2	51	2
3	4	54	1
3	5	57	2
4	0	61	0
4	1	63	1
4	2	66	2
4	3	69	2
4	5	72	0
5	0	75	1
5	1	78	2
5	2	81	0
5	3	84	1
5	4	87	0

Table 5: Toy problem RWA solutions

References

[1] Ethan Davies et al. "Optical Routing with Binary Optimisation and Quantum Annealing". In: *arXiv preprint arXiv:2402.07600* (2024).

[2] Aleksey S Boev et al. "Quantum-inspired optimization for wavelength assignment". In: *Frontiers in Physics* 10 (2023), p. 1092065.

[3] Charles A. Brackett. "Dense wavelength division multiplexing networks: Principles and applications". In: *IEEE Journal on Selected areas in Communications* 8.6 (1990), pp. 948–964.

[4] A Sangeetha et al. "Wavelength assignment problem in optical WDM networks". In: *International Journal of Recent Trends in Engineering* 1.3 (2009), p. 201.

[5] Rajiv Ramaswami and Kumar N Sivarajan. "Routing and wavelength assignment in all-optical networks". In: *IEEE/ACM Transactions on networking* 3.5 (1995), pp. 489–500.

[6] Gibong Jeong and Ender Ayanoglu. "Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks". In: *Proceedings of IEEE INFOCOM'96. Conference on Computer Communications*. Vol. 1. IEEE. 1996, pp. 156–163.

[7] Joseph Culberson. "Iterated greedy graph coloring and the difficulty landscape". In: (1992).

[8] Frank P Kelly. "Network routing". In: *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* 337.1647 (1991), pp. 343–367.

[9] Bowen Chen et al. "Multi-link failure restoration with dynamic load balancing in spectrum-elastic optical path networks". In: *Optical Fiber Technology* 18.1 (2012), pp. 21–28.

[10] Rajneesh Randhawa and JS Sohal. "Static and dynamic routing and wavelength assignment algorithms for future transport networks". In: *Optik* 121.8 (2010), pp. 702–710.

[11] Stephen G Brush. "History of the Lenz-Ising model". In: *Reviews of modern physics* 39.4 (1967), p. 883.

[12] Ronnie Kosloff. "Quantum thermodynamics: A dynamical viewpoint". In: *Entropy* 15.6 (2013), pp. 2100–2128.

[13] Adrian Bejan. "Evolution in thermodynamics". In: *Applied Physics Reviews* 4.1 (2017).

[14] Carla Silva et al. "Mapping graph coloring to quantum annealing". In: *Quantum Machine Intelligence* 2 (2020), pp. 1–19.

[15] Fred Glover, Gary Kochenberger, and Yu Du. "A tutorial on formulating and using QUBO models". In: *arXiv preprint arXiv:1811.11538* (2018).

[16] Andrew Lucas. "Ising formulations of many NP problems". In: *Frontiers in physics* 2 (2014), p. 5.

[17] Prasanna Date et al. "Efficiently embedding QUBO problems on adiabatic quantum computers". In: *Quantum Information Processing* 18 (2019), pp. 1–31.

[18] Egor S Tiunov, Alexander E Ulanov, and AI Lvovsky. "Annealing by simulating the coherent Ising machine". In: *Optics express* 27.7 (2019), pp. 10288–10295.

[19] Joseph Bowles et al. "Quadratic unconstrained binary optimization via quantum-inspired annealing". In: *Physical Review Applied* 18.3 (2022), p. 034016.

[20] Md Zahangir Alom et al. "Quadratic unconstrained binary optimization (QUBO) on neuromorphic computing system". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 3922–3929.

[21] Peter L Hammer and Eliezer Shlifer. "Applications of pseudo-Boolean methods to economic problems". In: *Theory and decision* 1 (1971), pp. 296–308.

[22] Prasanna Date, Davis Arthur, and Lauren Pusey-Nazzaro. "QUBO formulations for training machine learning models". In: *Scientific reports* 11.1 (2021), p. 10029.

[23] Maximilian Schlosshauer. "Quantum decoherence". In: *Physics Reports* 831 (2019), pp. 1–57.

[24] He-Liang Huang et al. "Superconducting quantum computing: a review". In: *Science China Information Sciences* 63 (2020), pp. 1–32.

[25] Colin D Bruzewicz et al. "Trapped-ion quantum computing: Progress and challenges". In: *Applied Physics Reviews* 6.2 (2019).

[26] Sergei Slussarenko and Geoff J Pryde. "Photonic quantum information processing: A concise review". In: *Applied Physics Reviews* 6.4 (2019).

[27] Elijah Pelofske. "Comparing three generations of d-wave quantum annealers for minor embedded combinatorial optimization problems". In: *arXiv preprint arXiv:2301.03009* (2023).

[28] Hirotoshi Yasuoka. "Computational complexity of quadratic unconstrained binary optimization". In: *arXiv preprint arXiv:2109.10048* (2021).

[29] Daniel Rehfeldt, Thorsten Koch, and Yuji Shinano. "Faster exact solution of sparse MaxCut and QUBO problems". In: *Mathematical Programming Computation* 15.3 (2023), pp. 445–470.

[30] Luca Trevisan. "Combinatorial optimization: exact and approximate algorithms". In: *Stanford University* (2011).

[31] Hiroki Oshiyama and Masayuki Ohzeki. "Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization". In: *Scientific reports* 12.1 (2022), p. 2146.

[32] Peter Merz and Bernd Freisleben. "Greedy and local search heuristics for unconstrained binary quadratic programming". In: *Journal of heuristics* 8 (2002), pp. 197–213.

[33] Yang Wang and Jin-Kao Hao. "Metaheuristic algorithms". In: *The Quadratic Unconstrained Binary Optimization Problem: Theory, Algorithms, and Applications*. Springer, 2022, pp. 241–259.

[34] Jirayu Supasil, Poramet Pathumsoot, and Sujin Suwanna. "Simulation of implementable quantum-assisted genetic algorithm". In: *Journal of Physics: Conference Series*. Vol. 1719. 1. IOP Publishing. 2021, p. 012102.

[35] Saıd Hanafi et al. "Tabu search exploiting local optimality in binary optimization". In: *European Journal of Operational Research* 308.3 (2023), pp. 1037–1055.

[36] Hiroshi Kagawa et al. "Fully-pipelined architecture for simulated annealing-based QUBO solver on the FPGA". In: *2020 Eighth International Symposium on Computing and Networking (CANDAR)*. IEEE. 2020, pp. 39–48.

[37] Stefanie Zbinden et al. "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies". In: *International Conference on High Performance Computing*. Springer. 2020, pp. 187–206.

[38] Peter L McMahon et al. "A fully programmable 100-spin coherent Ising machine with all-to-all connections". In: *Science* 354.6312 (2016), pp. 614–617.

[39] Yoshihisa Yamamoto et al. "Coherent Ising machines—optical neural networks operating at the quantum limit". In: *npj Quantum Information* 3.1 (2017), p. 49.

[40] Trevor Lanting et al. "Entanglement in a quantum annealing processor". In: *Physical Review X* 4.2 (2014), p. 021041.

[41] Tameem Albash et al. “Reexamination of the evidence for entanglement in the D-Wave processor”. In: *arXiv preprint arXiv:1506.03539* (2015).

[42] Catherine McGeoch and Pau Farré. “The D-wave advantage system: An overview”. In: *D-Wave Systems Inc., Burnaby, BC, Canada, Tech. Rep* (2020).

[43] AS Boev et al. “Genome assembly using quantum and quantum-inspired annealing”. In: *Scientific Reports* 11.1 (2021), p. 13183.

[44] Sergey R Usmanov et al. “Quantum and quantum-inspired optimization for an in-core fuel management problem”. In: *Journal of Physics: Conference Series*. Vol. 2701. 1. IOP Publishing. 2024, p. 012031.

[45] Barry A Cipra. “The Ising model is NP-complete”. In: *SIAM News* 33.6 (2000), pp. 1–3.

[46] Atanu Rajak et al. “Quantum annealing: An overview”. In: *Philosophical Transactions of the Royal Society A* 381.2241 (2023), p. 20210417.

[47] Abhishek Shukla, Mitali Sisodia, and Anirban Pathak. “Complete characterization of the directly implementable quantum gates used in the IBM quantum processors”. In: *Physics Letters A* 384.18 (2020), p. 126387.

[48] David Hestenes. “Spin and uncertainty in the interpretation of quantum mechanics”. In: *American Journal of Physics* 47.5 (1979), pp. 399–415.

[49] Mohsen Razavy. *Quantum theory of tunneling*. World Scientific, 2013.

[50] Bikas K Chakrabarti and Arnab Das. “Transverse Ising model, glass and quantum annealing”. In: *Quantum Annealing and Other Optimization Methods* (2005), pp. 1–36.

[51] Theodore A Wertime. “The Beginnings of Metallurgy: A New Look: Arguments over diffusion and independent invention ignore the complex metallurgic crafts leading to iron.” In: *Science* 182.4115 (1973), pp. 875–887.

[52] Frederick John Humphreys and Max Hatherly. *Recrystallization and related annealing phenomena*. elsevier, 2012.

[53] Satoshi Morita and Hidetoshi Nishimori. “Mathematical foundation of quantum annealing”. In: *Journal of Mathematical Physics* 49.12 (2008).

[54] Tameem Albash and Daniel A Lidar. “Decoherence in adiabatic quantum computation”. In: *Physical Review A* 91.6 (2015), p. 062320.

[55] Travis S Humble et al. “Quantum computing circuits and devices”. In: *IEEE Design & Test* 36.3 (2019), pp. 69–94.

[56] Dennis Willsch et al. “Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems”. In: *Quantum Information Processing* 21.4 (2022), p. 141.

[57] Andrew D King et al. “Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets”. In: *Nature communications* 12.1 (2021), p. 1113.

[58] Tameem Albash and Daniel A Lidar. “Demonstration of a scaling advantage for a quantum annealer over simulated annealing”. In: *Physical Review X* 8.3 (2018), p. 031016.

[59] Catherine C McGeoch et al. “A comment on comparing optimization on D-Wave and IBM quantum processors”. In: *arXiv preprint arXiv:2406.19351* (2024).

[60] Feng Zhao et al. “Wavelength division multiplexers/demultiplexers for optical interconnects in massively parallel processing”. In: *Optical Engineering* 42.1 (2003), pp. 273–280.

[61] *Mixed-Integer Programming (MIP) – A Primer on the Basics - Gurobi Optimization — gurobi.com*. `https://www.gurobi.com/resources/mixed-integer-programming-mip-a-primer-on-the-basics/`. [Accessed 13-08-2024].

[62] Zhipeng Lü, Jin-Kao Hao, and Fred Glover. "A study of memetic search with multi-parent combination for UBQP". In: *Evolutionary Computation in Combinatorial Optimization: 10th European Conference, EvoCOP 2010, Istanbul, Turkey, April 7-9, 2010. Proceedings 10*. Springer. 2010, pp. 154–165.

[63] Dmitrii Beloborodov et al. "Reinforcement learning enhanced quantum-inspired algorithm for combinatorial optimization". In: *Machine Learning: Science and Technology* 2.2 (2020), p. 025009.