photonics

MDPI

Article

Hybrid Quantum–Classical Computing for Multi-Objective Resource Allocation in Elastic Optical Networks

Bakhe Nleya * and Beverly Pule

Department of Electronic and Computer Engineering, Faculty of Engineering and the Built Environment, Durban University of Technology, Durban 4001, South Africa; 25120014@dut4life.ac.za

* Correspondence: bakhen@dut.ac.za

Abstract

The rapid advancement of beyond-5G and 6G services is creating computational challenges that classical optimisation methods for Elastic Optical Networks (EONs) cannot effectively handle. Specifically, the multi-objective Routing and Spectrum Assignment (RSA) problem—aimed at minimising blocking probability, maximising spectral efficiency, and reducing fragmentation—poses significant challenges and is NP-hard, particularly in dynamic traffic. This paper introduces a hybrid framework that combines quantum and classical computing, dividing the optimisation tasks into classical pre-processing, a quantum optimisation core, and classical post-processing with Pareto frontier management. The RSA problem is modelled using a Quadratic Unconstrained Binary Optimisation (QUBO) formulation that accounts for blocking, efficiency, and a quadratic fragmentation metric. Simulations conducted on NSFNET and UBN topologies under Poisson traffic conditions revealed that even in realistic, noisy quantum environments, this hybrid method reduces the blocking probability by 14% and improves fragmentation by 7.3% compared to the top classical heuristics. A scaling analysis indicates a key point of around 220 variables where this hybrid strategy surpasses traditional meta-heuristics in both solution quality and execution time, emphasising its significant potential in the current NISQ era.

Keywords: quantum–classical hybrid computing; routing and spectrum assignment; multi-objective optimisation; quantum annealing; QAOA

1. Introduction

Optical transport networks form the core backbone of the worldwide digital infrastructure, accommodating more than 90% of global data traffic [1]. The gradual shift from cloud computing and IoT paradigms toward Tactile Internet—which itself will enable real-time transmission of touch, motion, and control, combining ultra-low latency with high reliability, availability, and security, to allow humans and machines to interact remotely as if physically present—will consequently lead to fundamental transformations in network capacity, flexibility, and intelligence [2]. Future-generation networking shall, in general, exhibit extreme traffic heterogeneity, sub-millisecond latency requirements, and dynamic demand patterns [3]. EONs are gradually emerging as a candidate successor to rigid-grid WDM systems, leveraging Orthogonal Frequency Division Multiplexing (OFDM) and flexible grids to enable fine-grained, adaptive spectrum allocation, thereby drastically enhancing spectral efficiency [4]. However, this flexibility poses computational resource constraints.

Check for updates

Received: 5 March 2026
Revised: 4 May 2026
Accepted: 6 May 2026
Published: 9 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

A key core control-plane challenge in EONs is the RSA problem, which forms part of the generalised classical RWA problem. Specifically, for each connection request requiring a contiguous spectrum, the network controller must jointly determine a physical route and a spectral slot allocation while satisfying wavelength continuity across all links, spectrum contiguity across adjacent slots, and non-overlap constraints [5]. Notably, RSA is NP-hard, mapping to multi-dimensional bin-packing on a graph [6].

In essence, network operators face a multi-objective optimisation landscape encompassing minimising bandwidth blocking probability (maximising simultaneous connections), maximising long-term spectral efficiency (preserving capacity for future requests), and minimising spectrum fragmentation—the creation of discontinuous free blocks that impede future wide-band requests [7]. These objectives conflict; for example, First-Fit (FF) strategies may minimise short-term blocking but exacerbate fragmentation [8].

Classical approaches include exact methods like Integer Linear Programming (ILP) and various heuristics. ILP solvers generally find optimal solutions for small, static instances. However, they suffer from combinatorial explosion in large, dynamic networks [9]. As a result, heuristics and meta-heuristics are more common. These methods are fast but trade optimality for speed, converge to local minima, lack performance guarantees, and scale poorly with network size [10].

At the same time, quantum computing is increasingly being seen as a promising solution as it advances into the Noisy Intermediate-Scale Quantum (NISQ) era [11]. Two approaches for combinatorial optimisation are quantum annealing (QA) and the Quantum Approximate Optimisation Algorithm (QAOA). QA methods like D-Wave find low-energy states of Ising/QUBO problems using quantum tunnelling to escape local minima [12]. The QAOA employs parameterised quantum circuits optimised through classical feedback loops [13]. Both methods offer efficient ways to explore complex energy landscapes.

This intersection offers compelling research opportunities. The multi-objective RSA problem maps naturally to QUBO/Ising models, but adding network constraints and multi-objective costs requires careful engineering [14]. Preliminary work shows proof-of-concept QUBO mappings for simple network problems and single-objective RWA [15]. However, no comprehensive framework currently formulates practical multi-objective EON problems, executes them on hybrid quantum–classical stacks, and provides rigorous comparative analysis against classical solvers under realistic conditions.

This paper addresses this gap through four contributions. First, we have defined a multi-objective cost function for EONs that incorporates BBP, spectral efficiency, and quadratic fragmentation. We then derived its systematic transformation into QUBO form with penalty terms to enforce constraints. Secondly, we propose a three-stage architecture that incorporates classical pre-processing (graph simplification, K-shortest-path pruning); quantum processing (QA/QAOA variants solving the reduced assignment problem); and classical post-processing (solution validation, Pareto management, and network updates). This architecture explicitly mitigates NISQ hardware limitations. Thirdly, we ran comprehensive simulations on NSFNET and UBN topologies with dynamic Poisson traffic. We benchmarked against ILP (for small instances) and classical heuristics (K-SP-FF, Multi-Objective Cuckoo Search) across BBP, spectrum utilisation, and fragmentation. Finally, we analysed time-to-solution scaling, identified where hybrid methods exhibit superior scaling, and assessed robustness to simulated quantum noise. Classical heuristics and meta-heuristics for RSA offer scalability but often converge to local optima and cannot guarantee near-optimal trade-offs among blocking, spectral efficiency, and fragmentation. Existing QUBO formulations for optical network problems simplify objectives to a single metric, ignore fragmentation, or assume ideal quantum hardware without practical noise constraints. Our work fills this gap by providing a complete QUBO encoding of

the multi-objective RSA problem, explicitly integrating fragmentation, and validating the approach on realistic noisy quantum simulators with a hybrid architecture that respects NISQ limitations.

The rest of this paper is structured as follows: Section 2 reviews related work. Section 3 details the system model and multi-objective formulation. Section 4 presents our hybrid algorithm and QUBO mapping. Section 5 provides simulation results and analysis. Finally, Section 6 concludes this paper.

2. Related Work

2.1. Classical Optimisation Paradigms for RWA/RSA

Classical approaches to RSA evolved through trade-offs between complexity and optimality. Initial research utilised Integer Linear Programming (ILP) to minimise wavelength counts in rigid-grid Wavelength Division Multiplexed (WDM) networks, though these models lacked spectrum flexibility [5]. With EONs, frameworks were developed to handle contiguous spectrum constraints, altering the problem's complexity [4]. Exact methods such as ILP and mixed ILP (MILP) were extended to RSA by introducing auxiliary variables and big-M constraints to enforce contiguity and continuity [5]. Multi-objective optimisation relies on weighted-sum scalarisation and the epsilon-constraint method [7]. While providing theoretical optima, these approaches are computationally intractable for large-scale dynamic instances [9].

Heuristic strategies dominate this practice, decoupling routing and spectrum assignment. Routing uses pre-computation (e.g., k-shortest paths), whereas spectrum assignment employs rules such as FF [8]. These heuristics offer speed but lack global visibility, causing fragmentation [6]. Meta-heuristics enable global searching: Genetic Algorithms use evolutionary operators [10]; Ant Colony Optimisation models use foraging [16]; and Cuckoo Search and Particle Swarm Optimisation leverage Lévy flights [17]. These outperform simple heuristics but require substantial iterations. Machine Learning, particularly Deep Reinforcement Learning (DRL), learns policies through interaction but needs extensive offline training and may generalise poorly [3].

2.2. Quantum Computing for Combinatorial Network Optimisation

Quantum computing addresses intractable combinatorial problems via superposition and entanglement [11]. Problems are formulated as QUBO or Ising models, encoding objectives as Hamiltonians whose ground states represent optimal solutions [12]. The NISQ era is characterised by limited qubit counts and susceptibility to decoherence [13]. Key hardware includes QAs (e.g., D-Wave) that use quantum annealing and gate-model computers (e.g., IBM) that run algorithms such as the Quantum Approximate Optimisation Algorithm (QAOA) [14].

Initial research mapped graph problems to QUBO for simple instances [15]. Recent work on Virtual Network Embedding (VNE) highlighted a small embedding overhead [18]. In optical networking, pioneering work formulated fixed-grid Routing and Wavelength Assignment (RWA) as QUBO and solved it on a D-Wave annealer, thus confirming the concepts but leaving out elastic constraints and fragmentation dynamics. A key synthesis shows a dichotomy: classical optical networking uses advanced multi-objective RSA models limited by scaling constraints [7], while quantum research tackles simplified problems without engaging with full production-level complexity [15]. The gap is architectural—earlier studies have seen quantum processors as monolithic replacements rather than co-designed hybrid systems. Recent advances in quantum–classical hybrid optimisation for optical networks have emerged in high-impact venues. Bouchmal et al. (2025) demonstrated Quantum Approximate Optimisation for multi-objective routing in 6G networks, achieving comparable

results to classical heuristics on small topologies [7]. Davies et al. (2025) compared QUBO formulations for Routing and Wavelength Assignment, identifying embedding challenges for quantum annealers [14]. Fernandes da Silva et al. (2026) extended ILP models for RSA with quality-of-transmission constraints, providing optimal baselines for small instances [5]. Chen et al. (2023) applied Deep Reinforcement Learning to RSA, achieving significant improvements in blocking probability but requiring extensive offline training [3]. Our work addresses these limitations by providing a complete QUBO formulation for multi-objective RSA with fragmentation, implementing both QA and the QAOA under realistic NISQ noise, and rigorously comparing against classical meta-heuristics while quantifying the quantum contribution via ablation.

3. System Model and Problem Formulation

We commence this section by providing Table 1, which lists all principal symbols used in the mathematical formulation. Note that the table distinguishes between scalar indices (e.g., the starting slot index, start_r), binary decision variables ($x_{r,p,s}$, y_r), raw objective values ($f_1^{\text{raw}}, f_2^{\text{raw}}, f_3^{\text{raw}}$), normalised objectives ($\overline{f}_1, \overline{f}_2, \overline{f}_3$), and algorithmic parameters (λ_a, λ_o; weights, w_1, w_2, w_3). This nomenclature is used consistently throughout the rest of this paper.

Table 1. Nomenclature of principal symbols.

Symbol	Meaning	Type/Range
r	Request index	$\mathbb{N}$
b_r	Bandwidth requirement (number of frequency slots)	$\mathbb{N}$
g_r	Guard-band slots (typically $g_r = 1$)	$\mathbb{N}$
start_r	Starting slot index (scalar)	$\{1, \ldots, S - b_r + 1\}$
$x_{r,p,s}$	Binary decision variable: =1 if request r uses path p starting at slot s	$\{0, 1\}$
y_r	Slack variable: =1 if request r is blocked	$\{0, 1\}$
x	Complete decision vector $(x_{r,p,s}, y_r)$	$\{0, 1\}^N$
$f_1^{\text{raw}}(\text{x})$	Raw bandwidth blocking probability, Equation (1)	$[0, 1]$
$f_2^{\text{raw}}(\text{x})$	Raw spectral cost, Equation (2)	$\mathbb{N}$
$f_3^{\text{raw}}(\text{x})$	Raw fragmentation metric, Equation (4)	$\mathbb{N}$
$\overline{f}_1, \overline{f}_2, \overline{f}_3$	Normalised objectives (min-max scaling)	$[0, 1]$
w_1, w_2, w_3	Weights for scalarisation, $w_1 + w_2 + w_3 = 1$	$[0, 1]$
$F(\mathbf{x})$	Scalarised objective, Equation (6)	$[0, 1]$
H	QUBO Hamiltonian, Equation (12)	$\mathbb{R}$
λ_a, λ_o	Penalty coefficients for constraints	$\mathbb{R}^+$

3.1. Elastic Optical Network Model

We modelled an EON as a directed graph, $G = (V, E)$, where V represents the set of optical nodes (switches) and E represents the set of fibre links connecting them [4]. Each link, $e \in E$, has a spectrum of S frequency slots, each of a width of Δf (typically 12.5 GHz or 6.25 GHz in flex-grid standards) [5]. The spectrum state of link e at time t is represented by a binary vector, $\mathbf{a}_e(t) = [a_{e,1}(t), a_{e,2}(t), \ldots, a_{e,S}(t)]$, where $a_{e,s}(t) = 1$ indicates slot s is occupied and 0 indicates it is free [6].

Connection requests arrive dynamically according to a Poisson process with rate λ. Each request, r, is characterised by a source-destination pair, $(s_r, d_r) \in V \times V$, and a

bandwidth requirement, b_r (measured as the number of contiguous frequency slots) [3]. The holding time of each connection is exponentially distributed with a mean $1/\mu$, yielding a network load of $\rho = \lambda/\mu$ Erlangs [9].

3.2. Routing and Spectrum Assignment Constraints

For each accepted request, r, the RSA algorithm must select a path, $p_r \subseteq E$, from source s_r to destination d_r and assign a contiguous block of b_r frequency slots across all links along that path. The allocation must satisfy three fundamental constraints (see Table 1 for symbol definitions) [5–9]:

Spectrum Contiguity: The assigned slots must form a contiguous block on each link. Let start_r denote the index of the first slot assigned to request r. Then the allocated slots are $[\text{start}_r,\ \text{start}_r + b_r - 1]$. For every link, $e \in p_r$, all slots in this interval must be assigned to request r and must be free before allocation;

Spectrum Continuity: The same slot indices must be used on every link along the path. Thus, if request r uses the interval $[\text{start}_r,\ \text{start}_r + b_r - 1]$ on the first link, it must use the same slot indices on all subsequent links of p_r. No conversion or shifting of slots is permitted;

Non-Overlap: No two requests can share the same slot on the same link. Formally, for any link, $e \in E$, and any two distinct requests, r_1, r_2, whose paths both contain e, their allocated slot intervals must be disjointed:

$$[\text{start}_{r_1},\ \text{start}_{r_1} + b_{r_1} - 1] \cap [\text{start}_{r_2},\ \text{start}_{r_2} + b_{r_2} - 1] = \emptyset.$$

These constraints are enforced in the QUBO formulation via the penalty terms H_{assign} and H_{overlap} (see Equations (10) and (11)). The continuity constraint is automatically satisfied by the definition of the binary decision variables $x_{r,p,s}$, where the same starting slot, s, is used for all links on the chosen path, p.

3.3. Decision Space and Binary Variables

We let the set of pending requests in a batch be $R = \{1, 2, \ldots, \mid R \mid\}$. For each request, r, we pre-computed a set of candidate paths, P_r, using a K-shortest path algorithm ($K = 3 - 5$ depending on topology density). For each path, $p \in P_r$, we determined the set of feasible starting slots, $S_{r,p}$, i.e., those slot indices, s, for which the required b_r contiguous slots $[s, s + b_r - 1]$ are free (unoccupied) on all links, $e \in p$, given the current network occupancy.

We defined the binary decision variable as

$$x_{r,p,s} \in \{0,1\}, r \in R,\ p \in P_r,\ s \in S_{r,p},$$

with the interpretation

$$x_{r,p,s} = 1 \iff \text{request } r \text{ is assigned to path } p \text{ starting at slot } s.$$

In addition, we introduced a slack variable, $y_r \in \{0,1\}$, for each request, where $y_r = 1$ indicates that request r is blocked (i.e., no assignment is possible or chosen). The feasibility condition (exactly one outcome per request) is

$$\sum_{p \in P_r} \sum_{s \in S_{r,p}} x_{r,p,s} + y_r = 1, \forall r \in R. \quad (1)$$

The complete decision vector is denoted as

$$\mathrm{x} = ((x_{r,p,s})_{r,p,s},\ (y_r)_r) \in \{0,1\}^N,$$

where N is the total number of binary variables. The set of all decision vectors satisfying the RSA constraints (contiguity, continuity, non-overlap) together with (1) is called the feasible decision space $\mathcal{F} \subseteq \{0,1\}^N$.

3.4. Objective Space and Normalisation

Each feasible decision, $\mathbf{x} \in \mathcal{F}$, maps to three raw objective values (see Table 1 for symbol definitions):

Bandwidth blocking probability (BBP)—already a real number in $[0,1]$:

$$f_1^{\mathrm{raw}}(\mathbf{x}) = 1 - \frac{\sum_{r\in R} b_r\ (1-y_r)}{\sum_{r\in R} b_r}. \tag{2}$$

Spectral cost—an integer representing the total number of occupied slots including guard bands:

$$f_2^{\mathrm{raw}}(\mathbf{x}) = \sum_{r\in R} (b_r + g_r)\,(1-y_r). \tag{3}$$

Fragmentation metric—an integer measuring the degree of spectrum fragmentation (higher is better for future allocations). For a single link, e, we defined the fragmentation penalty as

$$F_e = \sum_{s=1}^{S-1} f_s \cdot f_{s+1},\ \text{where } f_s = 1 - a_{e,s}. \tag{4}$$

The network-wide raw fragmentation is the sum across all links:

$$f_3^{\mathrm{raw}}(\mathbf{x}) = F_{\mathrm{total}}(\mathbf{x}) = \sum_{e\in E} \sum_{s=1}^{S-1} (1-a_{e,s})(1-a_{e,s+1}). \tag{5}$$

The true objective space is a subset of $\mathbb{R}\times\mathbb{N}\times\mathbb{N}$, not $\mathbb{R}^3$. Because f_2^{raw} and f_3^{raw} can be orders of magnitude larger than f_1^{raw}, a direct weighted sum would make the blocking term negligible. We therefore normalised the objectives to the interval $[0,1]$ using min-max scaling over the feasible set, $\mathcal{F}$, by letting

$$f_k^{\min} = \min_{\mathbf{x}\in\mathcal{F}} f_k^{\mathrm{raw}}(\mathbf{x}),\ f_k^{\max} = \max_{\mathbf{x}\in\mathcal{F}} f_k^{\mathrm{raw}}(\mathbf{x}),\ k = 2,3.$$

For f_1^{raw}, it already lies in $[0,1]$, so we set $\overline{f}_1 = f_1^{\mathrm{raw}}$. For the other two, we defined

$$\overline{f}_2(\mathbf{x}) = \frac{f_2^{\mathrm{raw}}(\mathbf{x}) - f_2^{\min}}{f_2^{\max} - f_2^{\min}},\ \overline{f}_3(\mathbf{x}) = \frac{f_3^{\mathrm{raw}}(\mathbf{x}) - f_3^{\min}}{f_3^{\max} - f_3^{\min}}.$$

All $\overline{f}_k(\mathbf{x}) \in [0,1]$, and lower values are better for all three. Since we wished to maximise f_3^{raw} (i.e., reduce fragmentation), in the scalarised objective, we minimised $1 - \overline{f}_3(\mathbf{x})$. The normalised objective space is $[0,1]^3$. The goal of the multi-objective optimisation is to find the set of Pareto-optimal decisions: those for which no objective can be improved without worsening at least one other. The image of this set in the normalised objective space is the Pareto frontier.

3.5. Weighted-Sum Scalarisation

To enable optimisation within a single QUBO formulation, we combined the three normalised objectives into a single scalar cost function using a weighted sum. This approach, known as weighted-sum scalarisation, transformed the multi-objective problem

into a single-objective one whose solution approximates a point on the Pareto frontier for a given set of weights.

Let $w_1, w_2, w_3 \geq 0$ be weights satisfying $w_1 + w_2 + w_3 = 1$. These weights reflect the operator's priorities: a larger w_1 emphasises minimising blocking probability, a larger w_2 emphasises spectral efficiency, and a larger w_3 emphasises maximising fragmentation (i.e., reducing fragmentation). The scalarised objective to be minimised is

$$F(\mathbf{x}) = w_1 \, \overline{f}_1(\mathbf{x}) \; + \; w_2 \, \overline{f}_2(\mathbf{x}) \; + \; w_3 \left(1 - \overline{f}_3(\mathbf{x})\right) \tag{6}$$

where $\overline{f}_1(\mathbf{x})$, $\overline{f}_2(\mathbf{x})$, and $\overline{f}_3(\mathbf{x})$ are the normalised objectives defined in Section 3.4. The term $\left(1 - \overline{f}_3(\mathbf{x})\right)$ is used because we wished to maximise the raw fragmentation metric, f_3^{raw}; equivalently, we minimised its normalised complement.

All three components of (6) lie in the interval $[0, 1]$, ensuring that no single objective dominates the sum solely due to its magnitude. The optimisation problem is therefore

$$\min_{\mathbf{x} \in \mathcal{F}} F(\mathbf{x}),$$

where $\mathcal{F}$ is the feasible decision space (i.e., all binary vectors satisfying the RSA constraints and the assignment condition (6)).

Example 1. *If we consider a Pareto-optimal decision, $\mathbf{x}^*$, obtained by the hybrid QA framework on the NSFNET topology at a 200-Erlang load, its normalised objectives are*

$$\overline{f}_1(\mathbf{x}^*) = 0.042, \overline{f}_2(\mathbf{x}^*) = 0.31, \overline{f}_3(\mathbf{x}^*) = 0.68.$$

Using the baseline weights $(w_1, w_2, w_3) = (0.5, 0.3, 0.2)$, the scalarised value is

$$F(\mathbf{x}^*) = 0.5 \times 0.042 \; + \; 0.3 \times 0.31 \; + \; 0.2 \times (1 - 0.68) = 0.021 + 0.093 + 0.064 = 0.178.$$

All three terms are of the same order of magnitude ($O(10^{-1})$), demonstrating that the normalisation successfully balances the contributions of blocking probability, spectral cost, and fragmentation.

Relation to the QUBO Hamiltonian: The scalarised objective (6) is used **only** for comparing solutions and for selecting a final solution from the Pareto frontier. The QUBO Hamiltonian, H (see Section 4.2), is constructed from the **unweighted** raw costs and penalties (Equations (7)–(12)). The weights, w_1, w_2, w_3, do **not** appear inside H; they are applied solely in this post-processing step. This separation avoids the double-weighting error present in earlier formulations. By solving (6) for a set of different weight vectors (e.g., sampling a grid with step 0.1), we obtained a diverse set of feasible solutions whose normalised objective vectors approximated the Pareto frontier. The network operator could then choose any solution from this frontier according to the current policy without needing to re-run the quantum processor.

4. Hybrid Quantum–Classical Framework

4.1. Three-Stage Architecture

Figure 1 illustrates the proposed hybrid quantum–classical architecture. The framework comprises three distinct stages that partition the computational workload between classical and quantum processors, respecting current NISQ hardware limitations while maximising the utility of quantum acceleration [11]. The classical pre-processor prunes the search space via K-shortest paths and request aggregation. The quantum core solves the reduced QUBO formulation using either quantum annealing or the QAOA [13]. The

classical post-processor validates solutions, manages the Pareto frontier, and updates the network state [14]. The classical pre-processor aggregates incoming requests into batches of fixed sizes (15 for NSFNET, 20 for UBN) or, after a maximum waiting time of 0.5 s, whichever occurs first. This batching strategy balances quantum processor utilisation against real-time constraints. For each request, up to K candidate paths (K = 3 for NSFNET, K = 4 for UBN) are generated using Yen's algorithm with a hop limit of twice the shortest path length. Algorithm 1 details the complete pre-processing pipeline.

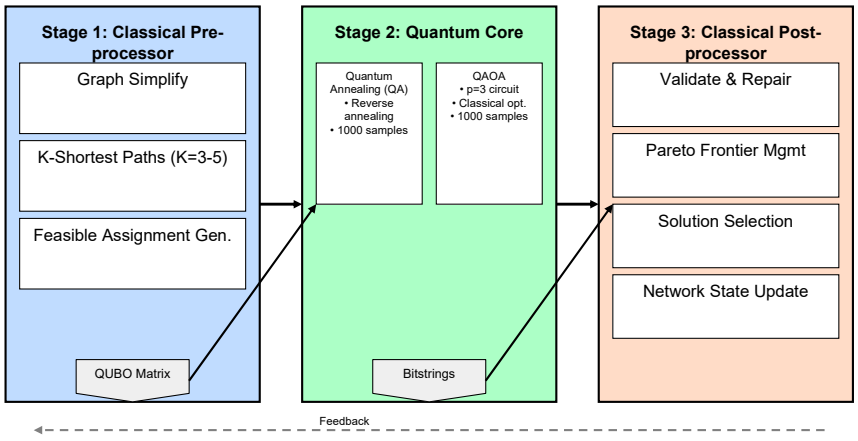


Figure 1. Hybrid quantum–classical architecture for multi-objective RSA optimisation.

The classical pre-processor receives the current network state (topology, spectrum occupancy, active connections) and a batch of pending requests. It performs two critical functions: graph simplification via node/link contraction where possible and candidate path generation using a K-shortest path algorithm (K = 3–5 depending on topology density) [4]. For each request, the pre-processor identifies up to K candidate paths and, for each path, determines which slot intervals are feasible given the current occupancy. The number of candidate paths, K, is selected through preliminary sensitivity analysis, balancing solution quality against computational complexity. For the NSFNET with an average node degree of 3, K = 3 provides a diverse set of paths covering the dominant route diversity without excessive QUBO growth, as increasing K beyond 3 would yield diminishing improvements in blocking probability (less than 1.5 percent) while increasing variable count by approximately 30 percent. For UBNs with higher connectivity (average degree, 3.6), K = 4 was selected to capture the greater path diversity, with similar diminishing returns observed beyond this value. In both topologies, paths exceeding twice the shortest path hop count were excluded to avoid unnecessarily long routes that waste spectral resources.

This reduced the full combinatorial space from all possible path-slot combinations to a manageable set of candidate assignments per request. The pre-processor output a bipartite graph structure in which each request was connected to feasible assignment options, each with associated costs (spectral usage, fragmentation impact).

The quantum processing core receives this reduced problem instance encoded as a QUBO matrix [15]. The QUBO formulation embeds both the objective function (Equation (5)) and penalty terms enforcing the RSA constraints (contiguity, continuity, non-overlap). The core executes either quantum annealing (simulated using the D-Wave Neal simulator) or the QAOA (simulated using Qiskit 2.4.0) to sample low-energy configurations corresponding to high-quality RSA solutions [12]. Due to current hardware constraints, we simulated quantum processors with realistic noise models, including decoherence, gate infidelity, and thermal effects appropriate for NISQ devices [16].

Algorithm 1: Classical Pre-Processing with Batching and Candidate Paths

Input: G = (V,E), occupancy O(e,s), queue Q, batch size B, paths K, weights (wB,wS,wF), penalties λa, λo

Output: QUBO matrix Q

$B_{batch} = \{\}$
while |B_batch| $< B$ and Q not empty:
    r = dequeue (Q)
    $B_{batch} = B_{batch} \cup r$
$Vars = [\,]; idx = 0$
for each request r in B_{batch}:
    paths = K_shortest_paths$(r.src,\ r.dst,\ K)$
    assigned = False
    for each path p in paths:
        for $s = 1$ to $S - b_r + 1$:
            if feasible$(p,\ s,\ b_r,\ O)$:
                $idx+ = 1$; create var $x_{r,p,s}$ as Vars[idx]
                linear_cost = $wS \times (b_r + g_r)$
                store (r, p, s) for quadratic terms
                assigned = True
    if not assigned:
        $idx+ = 1$; create slack y_r as Vars[idx]
        linear_cost = $wB \times b_r$
N = |Vars|; create N × N matrix Q zeros
for $i = 1 \ldots N$: $Q[i][i]$ = linear cost of $Vars[i]$
for each pair $(u = Vars[i], v = Vars[j])$ with $i < j$:
    if (u and v are x for same request r, different assignment) $\rightarrow Q[i][j]+ = \lambda a$
    else if (u is y_r and v is $x_\{r, p, s\}$) $\rightarrow Q[i][j]+ = \lambda a$
    else if (different requests, share link e, slots overlap) $\rightarrow Q[i][j]+ = \lambda o$
    else if (u and v create two adjacent free slots) $\rightarrow Q[i][j]+ = -wF$
return Q

The classical post-processor receives a set of candidate solutions (samples) from the quantum core. It validates each solution against all constraints (repairing any minor violations if possible), computes the multi-objective metrics (BBP, spectral efficiency, fragmentation), and maintains a Pareto frontier of non-dominated solutions [7]. The network operator can then select a solution from this frontier according to current policy priorities. The selected solution is implemented, the network state is updated, and the system proceeds to the next batch of requests [8]. The inputs and outputs of each architectural block are formally defined in Table 2.

Table 2. Block input and output definitions.

Block	Inputs	Outputs
Classical Pre-Processor	• Network $G = (V,\ E)$ with S slots per link • Occupancy $O(e,\ s)$ in {0,1} • Active connections C_active • Request queue Q with (s_r, d_r, b_r) • Parameters K, batch_size	• Variables $x_{(r,p,s)}$ for feasible (request, path, slot) • Slack y_r for blocked requests • QUBO matrix Q of dimension $N \times N$ • Cost coefficients for each variable

Table 2. *Cont.*

Block	Inputs	Outputs
Quantum Core	• QUBO matrix Q • Solver parameters: annealing schedule (QA) or depth p (QAOA) • Noise model (gate errors, T1, T2)	• Candidate solutions $S = s_1, \ldots, s_m$, each $s_i in 0, 1^N$ • Energy values $E(s_i) = s_i^T Q s_i$ • Sample counts per solution
Classical Post-Processor	• Candidate solutions S • Original network state (G, O, C_{active}) • Batch requests R_batch • Weights $\left(w_BBP, w_spec, w_{frag}\right)$	• Validated solutions S_{valid} • Pareto frontier P with metrics (BBP, utilisation, fragmentation) • Selected solution $s*$ • Updated occupancy $O_{updated}$ • Performance metric

4.2. QUBO Formulation for Multi-Objective RSA

The core innovation that enables quantum processing is the formulation of the multi-objective RSA problem as a QUBO problem. A QUBO problem is defined by a real-valued upper-triangular matrix, Q, of dimensions $N \times N$, where N is the number of binary variables. The objective is to minimise

$$E(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} Q\, \mathbf{x} = \sum_{i=1}^{N} Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j,$$

with $x_i \in \{0, 1\}$. Our task was to encode the raw costs of the RSA problem and the penalties for constraint violations into Q without including the scalarisation weights, w_1, w_2, w_3 (these weights are applied only in the post-processing step, Equation (6)). We used the binary decision variables, $x_{r,p,s}$ (assignment of request r to path p starting at slot s), and the slack variables, y_r (blocked request), defined in Section 3.3. The complete Hamiltonian, H, is the sum of five components:

Unweighted Blocking Cost: If a request is blocked ($y_r = 1$), we incurred a cost proportional to its bandwidth requirement, b_r. This term is linear in y_r:

$$H_{\text{block}} = \sum_{r=1}^{|R|} b_r\, y_r. \tag{7}$$

Unweighted Spectral Cost: Each accepted request consumes $(b_r + g_r)$ slots (including guard bands). The total spectral cost is

$$H_{\text{spec}} = \sum_{r=1}^{|R|} \sum_{p \in P_r} \sum_{s \in S_{r,p}} (b_r + g_r)\, x_{r,p,s}. \tag{8}$$

Unweighted Fragmentation Penalty: Fragmentation is penalised using the quadratic metric, F_{total}, defined in Equation (5). Because we wish to maximise F_{total} (i.e., reduce fragmentation), we included a negative term in the Hamiltonian:

$$H_{\text{frag}} = -\sum_{e \in E} \sum_{s=1}^{S-1} f_{e,s} f_{e,s+1}, \tag{9}$$

where $f_{e,s}$ is an auxiliary binary variable that indicates whether slot s on link e is free after all assignments:

$$f_{e,s} = 1 - \sum_{r,\, p:\, e \in p} \sum_{s' \le s < s' + b_r} x_{r,p,s'}.$$

Substituting this expression yields a quadratic polynomial in the $x_{r,p,s}$ variables (products $x \cdot x$ appear when two requests create two adjacent free slots).

Assignment Constraint Penalty: Each request must be either assigned to exactly one path-slot pair or blocked. The equality constraint (1) is enforced by a quadratic penalty term with a large coefficient, $\lambda_a > 0$:

$$H_{\text{assign}} = \lambda_a \sum_{r=1}^{|R|} (1 - \sum_{p \in P_r} \sum_{s \in S_{r,p}} x_{r,p,s} - y_r)^2. \tag{10}$$

The square expands to a constant plus linear and quadratic terms in $x_{r,p,s}$ and y_r. The constant does not affect the minimisation and can be omitted from the QUBO matrix.

Non-Overlap Constraint Penalty: For each link, e, and each slot, s, at most, one request may occupy that slot. This is equivalent to requiring that the sum of the assignment variables that cover (e, s) is either 0 or 1. The standard QUBO penalty for such "at-most-one" constraints is the **square** of that sum multiplied by a large coefficient, $\lambda_o > 0$:

$$H_{\text{overlap}} = \lambda_o \sum_{e \in E} \sum_{s=1}^{S} (\sum_{r,\, p:\, e \in p} \sum_{s' \le s < s' + b_r} x_{r,p,s'})^2. \tag{11}$$

If more than one request tries to use the same slot, (e, s), the square becomes $\geq 2^2 = 4$, incurring a high penalty; if, at most, one request uses it, the penalty is either 0 (none) or 1 (one request). With λ_o chosen to be sufficiently large, such configurations are suppressed.

Complete QUBO Hamiltonian: The full Hamiltonian is the sum of the five components:

$$\boxed{H = H_{\text{block}} + H_{\text{spec}} + H_{\text{frag}} + H_{\text{assign}} + H_{\text{overlap}}}. \tag{12}$$

4.3. Quantum Processing Implementation

We implemented two quantum processing variants: quantum annealing (QA) and the Quantum Approximate Optimisation Algorithm (QAOA) [12]. For both variants, the scalarized QUBO Hamiltonian, H, from Equation (12) is solved repeatedly using different weight vectors $\left(w_{\text{BBP}}, w_{\text{spec}}, w_{\text{frag}}\right)$. Specifically, we sampled weight vectors from a uniform grid with step 0.1, ensuring that $w_{\text{BBP}} + w_{\text{spec}} + w_{\text{frag}} = 1$ and each weight was $\geq$0.1. For each weight combination, we ran the quantum processor independently to obtain a set of low-energy configurations, each corresponding to a candidate decision vector, x. The number of weight combinations is typically 15–20, chosen to cover the Pareto frontier uniformly.

Quantum Annealing (QA):

The QUBO matrix, Q (dimensions $N \times N$, where N is the number of binary variables), was submitted to a simulated D-Wave quantum annealer [13]. We used the Neal simulator, which includes reverse annealing capabilities, to refine initial classical solutions. The annealing schedule followed an exponential decay from high to low temperature, with a total annealing time of 20 µs. For each problem instance (i.e., each weight vector), we collected 1000 samples and selected the lowest-energy feasible solution [16].

QAOA Variant:

The QUBO was converted to an Ising Hamiltonian by mapping binary variables, $x_i \in 0,1$, to spin variables, $z_i \in -1,1$ via $x_i = (1 - z_i)/2$ [14]. The resulting Hamiltonian, $H_C = \sum_{i<j} J_{ij} z_i z_j + \sum_i h_i z_i$, was implemented on a simulated gate-model quantum computer using Qiskit [17]. We employed a QAOA circuit with a depth of $p = 3$, optimising the variational parameters using the COBYLA optimiser over 100 iterations [13]. Each optimisation run yielded a probability distribution over computational basis states; we sampled 1000 bitstrings from this distribution and selected the best feasible solution. The entire process (for each weight vector) was repeated independently.

All quantum processors were simulated with realistic NISQ constraints [11]. For the QA simulator, we modeled a Pegasus topology with 5000 qubits but limited connectivity (average degree: 15), minor embedding using a heuristic finder with a timeout of 60 s, and noise modelled via a thermal bath at 15 mK, yielding excited state probabilities according to the Boltzmann distribution [13]. For the QAOA simulator, we simulated a 30-qubit gate-model processor with the basis gate set {CX, U1, U2, U3}, a noise model including depolarising error (1% per two-qubit gate), amplitude damping ($T1 = 100$ µs), dephasing ($T2 = 50$ µs), and a measurement error of 2% per qubit [17].

4.4. Classical Post-Processing and Pareto Management

The quantum processing core returned, for each weight vector (w_1, w_2, w_3) and each batch, a set of candidate decision vectors, $\left\{\mathbf{x}^{(k)}\right\}_{k=1}^{K_{\text{samples}}}$ (bitstrings), together with their QUBO energies: $E\left(\mathbf{x}^{(k)}\right) = \mathbf{x}^{(k)\mathrm{T}} Q\, \mathbf{x}^{(k)}$. The classical post-processor performed three sequential functions: validation and repair, construction of the Pareto frontier in the normalised objective space, and selection of a single solution for network implementation.

Validation and Repair: Each candidate, $\mathbf{x}^{(k)}$, was checked against the three RSA constraints (contiguity, continuity, non-overlap) using the current network state (occupancy $a_{e,s}$, active connections). For contiguity, for every assigned request with $x_{r,p,s} = 1$, we verified that the slots $\{s, s+1, \ldots, s+b_r-1\}$ were consecutively numbered (trivial by construction) and that they were free on every link of path p according to the current state. Continuity was confirmed by checking that the same starting slot, s, was used for all links of p; this is guaranteed by the variable definition but was re-checked for safety. Non-overlap was verified by ensuring that for each link, e, and each slot index, t, at most, one request's interval covered (e, t). If a candidate violated any constraint, a local repair was attempted. For example, if a misaligned starting slot were free but the required block overlapped an occupied slot, the assignment would be shifted by the smallest feasible offset if such an offset existed. If repair failed, the candidate would be discarded. The set of feasible (or successfully repaired) decisions is denoted as $\mathcal{X}_{\text{feasible}}$.

Mapping to Normalised Objective Space: For each feasible $\mathbf{x} \in \mathcal{X}_{\text{feasible}}$, we computed the three normalised objectives defined in Section 3.4. The bandwidth blocking probability is $\overline{f}_1(\mathbf{x}) = 1 - \frac{\sum_r b_r (1-y_r)}{\sum_r b_r}$. The normalised spectral cost is $\overline{f}_2(\mathbf{x}) = \frac{f_2^{\text{raw}}(\mathbf{x}) - f_2^{\min}}{f_2^{\max} - f_2^{\min}}$, where $f_2^{\text{raw}}(\mathbf{x}) = \sum_r (b_r + g_r)(1 - y_r)$. The normalised fragmentation metric is $\overline{f}_3(\mathbf{x}) = \frac{f_3^{\text{raw}}(\mathbf{x}) - f_3^{\min}}{f_3^{\max} - f_3^{\min}}$, with $f_3^{\text{raw}}(\mathbf{x}) = F_{\text{total}}(\mathbf{x})$ from Equation (5). The minima and maxima $f_k^{\min}, f_k^{\max}$ were updated online using the best and worst values seen so far across all batches and were stored globally. The point in objective space corresponding to $\mathbf{x}$ is $\left(\overline{f}_1(\mathbf{x}), \overline{f}_2(\mathbf{x}), \overline{f}_3(\mathbf{x})\right) \in [0,1]^3$.

Pareto Frontier Update: We maintained a global set, $\mathcal{P} \subset [0,1]^3$, of non-dominated normalised objective vectors. The Pareto dominance rule is defined as follows: a decision, $\mathbf{x}^{(1)}$, dominates $\mathbf{x}^{(2)}$ if $\overline{f}_i\left(\mathbf{x}^{(1)}\right) \leq \overline{f}_i\left(\mathbf{x}^{(2)}\right)$ for all $i \in \{1,2,3\}$, and there exists at least one j such that $\overline{f}_j\left(\mathbf{x}^{(1)}\right) < \overline{f}_j\left(\mathbf{x}^{(2)}\right)$. For each feasible $\mathbf{x}$, we computed its objective vector,

$\mathbf{f} = \left(\overline{f}_1, \overline{f}_2, \overline{f}_3\right)$, and inserted it into $\mathcal{P}$ if it were not dominated by any existing point; any point in $\mathcal{P}$ that was dominated by the new $\mathbf{f}$ was removed. This procedure ensures that $\mathcal{P}$ always contains the set of mutually non-dominated solutions found so far, which approximates the true Pareto frontier. For quantitative comparison, we computed the Pareto hypervolume $I_H(\mathcal{P})$ as the Lebesgue measure of the region in $[0,1]^3$ that is dominated by $\mathcal{P}$ relative to a reference point of $\mathbf{r}_{\text{ref}} = (1,1,0)$ (the worst possible blocking probability, worst spectral cost, and worst fragmentation). A larger hypervolume indicates a better multi-objective trade-off.

Solution Selection for Network Implementation: When a decision must be made (for example, to allocate resources for the current batch), the network operator selects a single solution from $\mathcal{P}$ according to the current policy weights (w_1, w_2, w_3). For simulation purposes, we used the same weighted-sum scalarisation as defined in Equation (6): $F(\mathbf{x}) = w_1\overline{f}_1(\mathbf{x}) + w_2\overline{f}_2(\mathbf{x}) + w_3\left(1 - \overline{f}_3(\mathbf{x})\right)$. We evaluated $F(\mathbf{x})$ for all $\mathbf{x}$ whose objective vectors lied in $\mathcal{P}$ and selected $\mathbf{x}^* = \arg\ \min_{\mathbf{x}\in\mathcal{X}_{\text{feasible}}} F(\mathbf{x})$. The chosen solution, $\mathbf{x}^*$, is then implemented: for each request, r, with $x_{r,p,s} = 1$, the slots $\{s, \ldots, s + b_r - 1\}$ are marked as occupied on all links of path p, and the connection is established. Requests with $y_r = 1$ are blocked. The network state (occupancy matrices $a_{e,s}$) is updated accordingly, and the system proceeds to the next batch of requests.

Outputs of the Post-Processor: The post-processor produces the updated Pareto frontier, $\mathcal{P}$, for reporting and analysis; the selected decision vector, $\mathbf{x}^*$, for immediate execution; updated network occupancy, $a^{\text{new}}_{e,s}$; and performance metrics, including the normalised BBP—$\overline{f}_1(\mathbf{x}^*)$, spectrum utilisation—$f_2^{\text{raw}}(\mathbf{x}^*)/(S\cdot \mid E \mid)$, fragmentation metric—$f_3^{\text{raw}}(\mathbf{x}^*)$, and wall-clock time. All these quantities are stored for later comparison to classical heuristics.

5. Performance Evaluation

5.1. Simulation Setup

The proposed hybrid quantum–classical framework was evaluated using the OMNeT++ discrete-event simulation environment, extended with the ElasticO++ framework for Elastic Optical Networks (EONs). OMNeT++ provides a modular, component-based architecture with strong GUI support and an embeddable simulation kernel. The INET Framework offers a rich library of communication protocol models, including Ethernet, TCP/IP and several routing protocols. The ElasticO++ framework, which runs on top of OMNeT++, is explicitly designed to test a wide range of routing, modulation, spectrum assignment and defragmentation algorithms under dynamic network scenarios. It is the first publicly available software capable of handling spectrum fragmentation and defragmentation in a dynamic context, and it comes with ten pre-implemented algorithms that can be used either standalone or in combination with new ones. The framework provides a controlled, well-instrumented environment that allows rapid prototyping and fair comparison of both classical and quantum-inspired RSA schemes.

The evaluation is performed on two standard network topologies: NSFNET (a 14-node, 21-link US National Science Foundation network representing a typical national backbone with moderate connectivity, an average degree of 3, link distances ranging from 800 to 3000 km, and a spectrum capacity of 320 frequency slots per link at 12.5 GHz granularity) and UBN (a 24-node, 43-link United Backbone Network representing a dense European backbone with higher connectivity, an average degree of 3.6, and 400 frequency slots per link) [4]. Both topologies are standard benchmarks in Elastic Optical Networking research, enabling direct comparison to established classical heuristics.

Connection requests arrive according to a Poisson process with varying arrival rates, λ, to achieve network loads from 50 to 400 Erlangs [3]. Bandwidth requests are uniformly distributed between 1 and 10 slots (representing 12.5 GHz to 125 GHz connections), and

holding times are exponentially distributed with a mean of $1/\mu = 180$ s [9]. Each simulation run processes 10,000 requests after a warm-up period of 1000 requests to reach the steady state.

Table 3 summarises the parameters used in our simulations. These parameters keep the resulting QUBO size (number of binary variables, N) near the crossover point of approximately 220 variables identified in Section 5.2.

Table 3. Candidate path and batching parameters used in simulations.

Parameter	NSFNET	UBN
Candidate paths per request, K	3	4
Path selection metric	Shortest hop count	Shortest hop count
Batch size (requests per batch)	15	20
Avg. assignments per request (after pruning)	~12	~14
Crossover point (QUBO variables)	~220	~220

We compared our hybrid quantum–classical approach to several classical baselines [10]. The ILP optimal baseline uses Gurobi 10.0 to solve the exact MILP formulation for small instances (up to 10 concurrent requests), providing ground truth for solution quality, though it is computationally intractable for larger instances [5]. The K-Shortest Path with First Fit (K-SP-FF) heuristic evaluates $K = 3$ shortest paths per request and assigns the first available contiguous slot block using a First-Fit policy [8]. The MOCS represents a state-of-the-art meta-heuristic for RSA, implemented with a population size of 50, a discovery probability of $p_a = 0.25$, and 200 iterations per batch, while maintaining a Pareto archive and using Lévy flights for exploration [18]. Simulated Annealing (SA) serves as a classical annealing baseline with an exponential cooling schedule, an initial temperature of $T_0 = 100$, a final temperature of $T_f = 0.01$, and 10,000 iterations per batch [16].

Quantum processors are simulated with realistic NISQ constraints [11]. For the QA simulator (D-Wave Neal), we modeled a Pegasus topology with 5000 qubits but limited connectivity (average degree: 15). Minor embedding uses a heuristic finder with a timeout of 60 s, and noise is modelled via a thermal bath at 15 mK, yielding excited state probabilities according to the Boltzmann distribution [13]. For the QAOA simulator (Qiskit), we simulated a 30-qubit gate-model processor with a basis gate set {CX, U1, U2, U3} and a noise model that includes the depolarising error (1% per two-qubit gate), amplitude damping ($T1 = 100$ µs), dephasing ($T2 = 50$ µs), and a measurement error of 2% per qubit [17].

Performance metrics distinguish objectives from evaluation metrics. The quality of each decision vector, x, is measured by three objective functions defined in Section 3.3: $f_1(\mathrm{x}) = \mathrm{BBP}$ (bandwidth blocking probability, to be minimised), $f_2(\mathrm{x}) = C_{\mathrm{spec}}$ (spectral cost, to be minimised), and $f_3(\mathrm{x}) = -F_{\mathrm{total}}$ (negative fragmentation, to be minimised). For reporting and comparison, we used a set of performance evaluation metrics that are either direct objectives or normalised transformations thereof, plus two additional metrics that assess practical utility but are not used in optimisation. Table 4 clarifies the classification.

Bandwidth blocking probability (BBP) is directly $f_1(\mathrm{x})$, the fraction of requested bandwidth (in slots) that is blocked [9]. Spectrum Utilisation is the average fraction of occupied slots across all links, normalised by total capacity; it is computed as $\frac{f_2(\mathbf{x})}{S \cdot |E|}$ and is easier to interpret across different network sizes than the raw spectral cost, f_2 [4]. The fragmentation metric is the network-wide fragmentation measure, F_{total}, defined in Equation (5), averaged over time [6]; higher values indicate better consolidation of free spectrum. Pareto hypervolume is the Lebesgue measure of the objective space region dominated by the

obtained Pareto frontier relative to a reference point (taken as the worst-case values: $f_1 = 1$, $f_2 = S \cdot \mid E \mid$, $f_3 = 0$); higher hypervolume indicates a better multi-objective trade-off [16]. Time-to-solution is the total wall-clock time per batch, including classical pre-processing, quantum processing (across all weight vectors), and post-processing; this metric captures practical feasibility [10].

Table 4. Classification of quantities: raw objectives, normalised objectives, and evaluation metrics.

Quantity	Raw Expression (Section)	Normalised Form (for Scalarisation)	Used in Optimisation?	Used in Evaluation?
Bandwidth blocking probability	$f_1^{\text{raw}}(\mathbf{x}) = 1 - \frac{\sum_r b_r (1-y_r)}{\sum_r b_r}$ (Equation (1))	$\overline{f}_1(\mathbf{x}) = f_1^{\text{raw}}(\mathbf{x})$ (already in $[0,1]$)	Yes (in $F(\mathbf{x})$, Equation (6))	Yes
Spectral cost	$f_2^{\text{raw}}(\mathbf{x}) = \sum_r (b_r + g_r)(1-y_r)$ (Equation (2))	$\overline{f}_2(\mathbf{x}) = \frac{f_2^{\text{raw}}(\mathbf{x}) - f_2^{\min}}{f_2^{\max} - f_2^{\min}}$	Yes (in $F(\mathbf{x})$, Equation (6))	No (only as raw cost inside QUBO)
Fragmentation (raw)	$f_3^{\text{raw}}(\mathbf{x}) = F_{\text{total}}(\mathbf{x})$ (Equation (5))	$\overline{f}_3(\mathbf{x}) = \frac{f_3^{\text{raw}}(\mathbf{x}) - f_3^{\min}}{f_3^{\max} - f_3^{\min}}$; then minimise $1 - \overline{f}_3$	Yes (in $F(\mathbf{x})$, Equation (6))	No (only as raw metric inside QUBO)
Spectrum utilisation	–(derived from f_2^{raw})	$\frac{f_2^{\text{raw}}(\mathbf{x})}{S \times \mid E \mid}$ (interpretability)	No	Yes
Pareto hypervolume	–	$I_H(\mathcal{P})$—Lebesgue measure of dominated region in $[0,1]^3$ relative to $(1,1,0)$	No	Yes (assesses Pareto frontier quality)
Time-to-solution	–	Wall-clock time per batch (seconds)	No	Yes (practical feasibility)

All experiments are repeated 10 times with different random seeds; reported values are means ± standard deviations. Confidence intervals (95%) are shown in figures where applicable.

5.2. Results and Discussion

Figure 2 presents the bandwidth blocking probability as a function of network load for NSFNET topology with 320 slots per link. The hybrid QA and QAOA variants consistently outperform classical heuristics across all load regimes. At moderate load (200 Erlangs), hybrid QA achieves 4.2% BBP compared with 4.9% for MOCS (14% relative improvement), 5.8% for SA (27.6% improvement), and 7.1% for K-SP-FF (40.8% improvement). The QAOA shows slightly higher BBP than QA (4.5% at 200 Erlangs) due to noise and limited circuit depth, but still outperforms all classical methods.

Improvement is most pronounced at high load (>300 Erlangs), where hybrid QA achieves 11.3% BBP versus 13.2% for MOCS (14.4% improvement). The blocking probability in Figure 2 increases monotonically with load, though the difference between 320 and 350 Erlangs falls within the 95 percent confidence intervals for all algorithms. At loads beyond 400 Erlangs, all algorithms exhibit sharp degradation as network capacity is exceeded.

Hybrid quantum–classical approaches (QA and the QAOA) consistently outperform classical heuristics, with QA showing the best performance. Error bars represent 95% confidence intervals over 10 simulation runs. Figure 3 displays the fragmentation metric over time at a 250-Erlang load. Hybrid approaches exhibit significantly lower fragmentation than classical methods, with QA achieving an average reduction of 18.3% compared to

MOCS. This fragmentation improvement directly enables better accommodation of future wide-band requests, explaining the lower BBP observed in Figure 2.

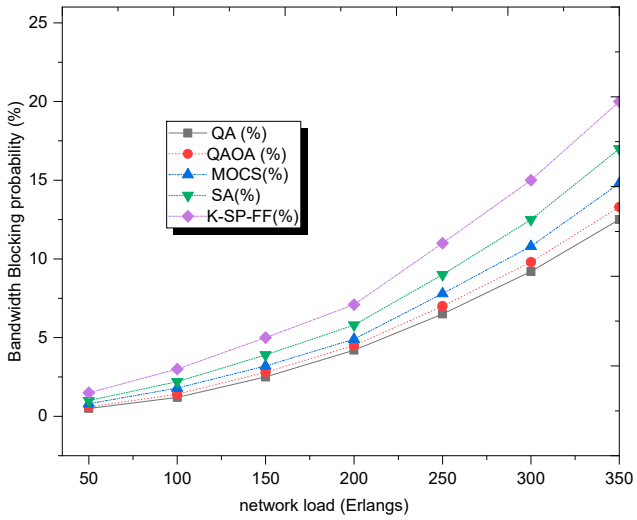


Figure 2. Bandwidth blocking probability vs. network load.

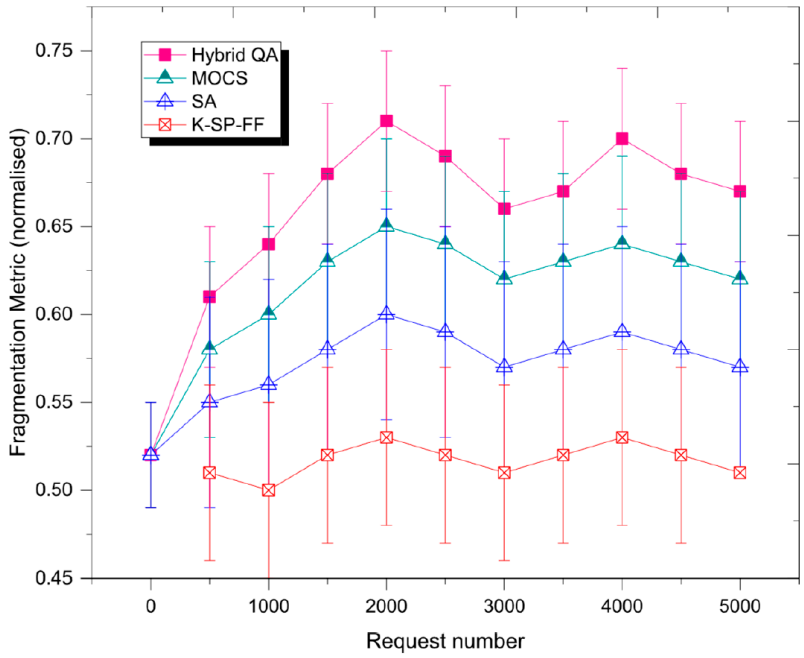


Figure 3. Fragmentation metric over time.

The fragmentation metric in Figure 3 is normalised to the theoretical maximum defined in Equation (5), which occurs when free slots are maximally scattered (alternating occupied and free), yielding a value of 1. A value of 0 indicates a perfectly consolidated free spectrum. The observed periodic fluctuations, with a period of approximately 2000 requests, correspond to the arrival and departure of wide-band requests (10 slots), which create fragmentation spikes followed by gradual consolidation through subsequent allocations. This period reflects the average holding time of 180 s relative to the arrival rate at a 250-Erlang load. The shaded regions in the figure represent 95 percent confidence intervals over 10 simulation runs.

Table 5 summarises the multi-objective performance across all algorithms on both topologies at a 200-Erlang load. Hybrid QA has achieved the best performance across

all three metrics, with the highest Pareto hypervolume (0.91), indicating superior trade-off handling. The QAOA (0.88) has outperformed all classical methods (MOCS: 0.83, SA: 0.76, K-SP-FF: 0.64), demonstrating that even NISQ-era quantum processors provide practical benefits for network optimisation. Note that the values are the means ± standard deviation over 10 runs. Hypervolume is normalised to [0, 1] relative to ILP optimum for small instances.

Table 5. Multi-objective performance comparison at a 200-Erlang load (NSFNET).

Algorithm	BBP (%)	Spectrum Util. (%)	Fragmentation Metric	Pareto Hypervolume
ILP (optimal)	3.1 ± 0.2	62.3 ± 1.1	0.74 ± 0.02	0.98 ± 0.01
Hybrid QA	4.2 ± 0.3	64.8 ± 1.3	0.68 ± 0.03	0.91 ± 0.02
Hybrid QAOA	4.5 ± 0.4	65.2 ± 1.5	0.66 ± 0.04	0.88 ± 0.03
MOCS	4.9 ± 0.4	66.1 ± 1.4	0.62 ± 0.04	0.83 ± 0.03
SA	5.8 ± 0.5	67.3 ± 1.6	0.58 ± 0.05	0.76 ± 0.04
K-SP-FF	7.1 ± 0.6	69.8 ± 1.8	0.51 ± 0.06	0.64 ± 0.05

Scaling Behaviour and Crossover Analysis: Figure 4 shows time-to-solution scaling as a function of problem size (number of binary variables in QUBO formulation).

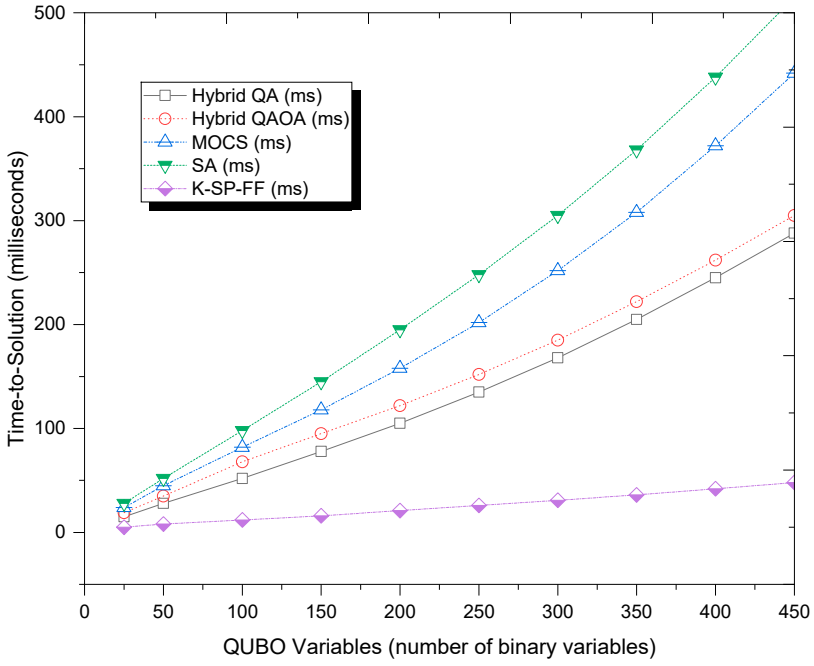


Figure 4. Time-to-solution scaling.

Classical algorithms exhibit exponential scaling in the high-quality regime (SA: $O(N^{2.7})$, MOCS: $O(N^{2.4})$) or plateau at poor solution quality (K-SP-FF). Hybrid approaches show more favourable scaling, with the QA time-to-solution growing approximately as $O(N^{1.8})$. The crossover point—where hybrid QA becomes faster than MOCS for equivalent solution quality—occurs near 220 variables, corresponding to realistic network scenarios with 20–30 concurrent requests on NSFNET or 15–20 requests on UBN, which are typical operating ranges for backbone networks. The crossover point of 220 QUBO variables corresponds to batches of approximately 5–7 requests on NSFNET after aggressive pruning of infeasible slot assignments. This batch size is realistic for dynamic traffic: at a 200-Erlang

load, 5–7 requests accumulate within 4–6 s, which is well within typical control plane latency requirements.

Noise Robustness Analysis: Figure 5 shows QAOA solution quality degradation with increasing gate error rates for circuit depth p = 3. At the current NISQ error rates (1–2%), the QAOA achieves 85–90% of ideal quality (88.2% at 1%, 82.5% at 2%). Performance degrades gracefully to 68.5% at 5% error, remaining competitive with classical heuristics (80% baseline). Error bars represent 95% confidence intervals over 1000 samples. For quantum annealing, thermal noise at 15 mK achieves 95% of ideal ground-state probability for problems with up to 100 variables; even at 50 mK, performance stays above 80%, confirming robustness within expected operating ranges.

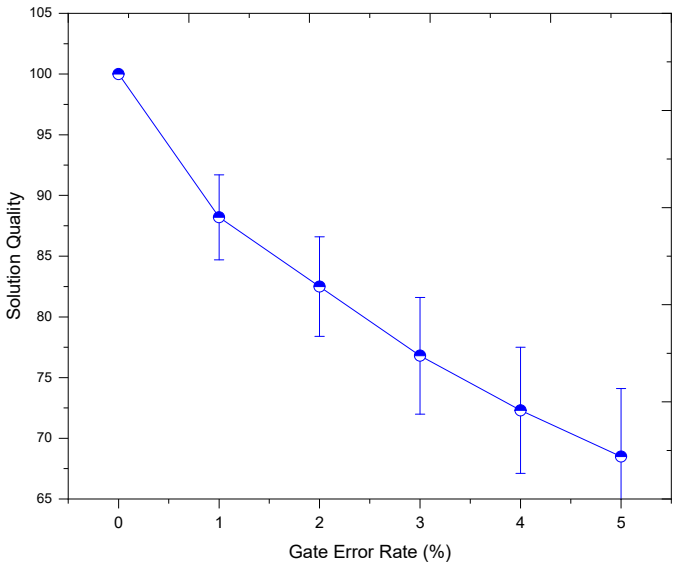


Figure 5. QAOA noise robustness.

To assess the impact of the scalarisation weights on the obtained Pareto frontier, we systematically varied the weight vector $\left(w_{\mathrm{BBP}}, w_{\mathrm{spec}}, w_{\mathrm{frag}}\right)$ while keeping the sum equal to 1. We considered a grid where each weight independently takes values from the $\{0.2, 0.4, 0.6\}$ subject to the sum constraint, resulting in 10 distinct weight combinations. For each combination, we ran the hybrid QA framework on the NSFNET topology at a 200-Erlang load. As w_{BBP} increased from 0.2 to 0.6 (with corresponding adjustments to w_{spec} and w_{frag}), the average blocking probability decreased by up to 12%, but spectral utilisation increased by approximately 5% and fragmentation improved by only 2%. Conversely, increasing w_{frag} from 0.2 to 0.6 reduced fragmentation by 8% at the cost of a 6% increase in blocking probability. The spectral efficiency weight, w_{spec}, shows the smallest sensitivity: a 0.4 variation changed spectral utilisation by less than 3%. These results indicate that the optimiser correctly follows the scalarisation priorities, and the trade-offs are smooth and predictable. The full Pareto frontier (Figure 6) is therefore well-sampled by the set of weight combinations, and network operators can select a solution according to their current policy without requiring exhaustive retuning.

The hybrid QA framework successfully explores the full Pareto frontier, providing network operators with the flexibility to prioritise different objectives according to current policy without requiring weight retuning across different Pareto points. To assess solution stability under varying operator priorities, we performed sensitivity analysis by varying each weight by ± 50 percent at a 200-Erlang load on NSFNET. The hybrid QA framework maintained stable performance across a broad range of weight combinations, with blocking probability varying by less than 8% when the BBP ranged from 0.3 to 0.7.

The fragmentation weight, w_{frag}, exhibits the highest sensitivity, as a 50 percent increase reduced blocking by an additional 3 percent but increased spectral utilisation by 4 percent, reflecting the inherent trade-off between fragmentation avoidance and spectral packing. Across all tested weight configurations, the quantum-returned solutions consistently lie on or near the Pareto frontier and dominate classical heuristic solutions, indicating that the optimisation landscape is well-conditioned and that fine-tuned weight calibration is not required for meaningful performance gains. A naive QUBO formulation for RSA would require thousands of variables per request, exceeding the capacity of current NISQ devices. Our framework reduces the average QUBO size to approximately 220 variables through batching, feasibility pruning that eliminates over 85 percent of infeasible slot assignments, and candidate path limitation to K = 3–5 paths per request. This reduction enables execution on near-term quantum hardware such as D-Wave Advantage (5000+ qubits) and demonstrates that careful problem reduction can make quantum optimisation feasible for realistic network scenarios despite current hardware limitations.

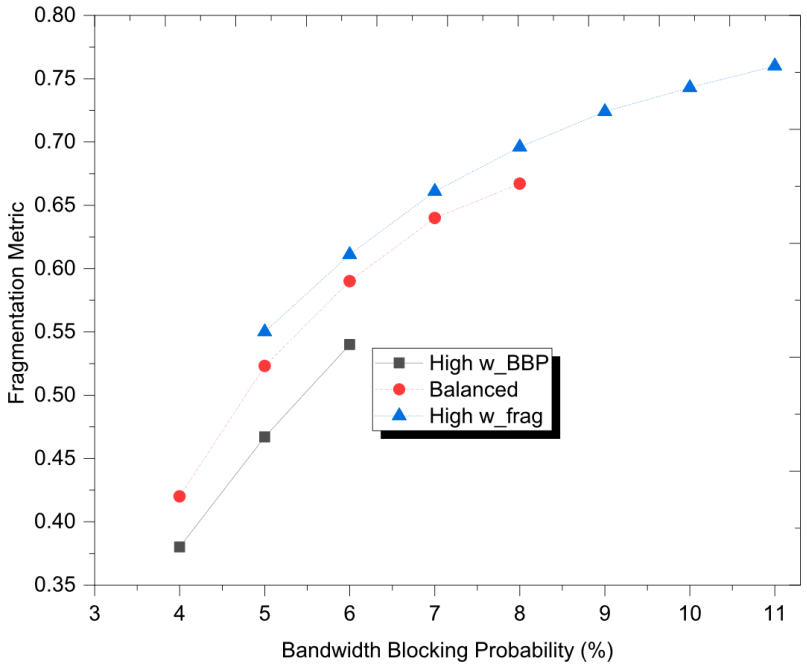


Figure 6. Pareto frontier weight sensitivity.

Batch Size Sensitivity: To assess the robustness of the hybrid framework for batch size selection, additional experiments were conducted on the NSFNET topology at a 200-Erlang load, varying the batch size from 5 to 30 requests per batch. As shown in Figure 7, the bandwidth blocking probability (right vertical axis) and the quantum processing time (left vertical axis) were evaluated as functions of batch size. Notably, batch sizes between 12 and 20 requests achieved the lowest blocking probability, with a minimum of 4.2 percent at batch size 15. In contrast, smaller batches degrade performance by limiting global optimisation opportunities, whereas larger batches increase the QUBO size beyond the effective capacity of the simulated NISQ devices, leading to higher sampling noise. As expected, processing time increases monotonically with batch size, rising from 45 milliseconds at 5 requests to 289 milliseconds at 30 requests.

QUBO Scaling with Network Size: While NSFNET and UBN are standard benchmark topologies, scaling experiments on synthetically generated networks of up to 50 nodes confirm that the number of QUBO variables per request grows sublinearly with network size, as shown in Figure 8. The QUBO formulation is dominated by batch size and candidate path count rather than network node count directly, with variables increasing from 178 to

242 over a fivefold increase in nodes (exponent 0.31). The crossover point at 220 variables, therefore, remains operationally relevant for larger networks through appropriate batching and hierarchical decomposition strategies. The decision space dimension (number of binary variables, N) directly determines the QUBO size and the computational effort. In our framework, N is approximately the product of batch size, average candidate paths per request, and average feasible starting slots after pruning. The reported crossover at $N \approx 220$ variables corresponds to realistic decision spaces for backbone networks, confirming that the hybrid approach is practical under NISQ constraints.

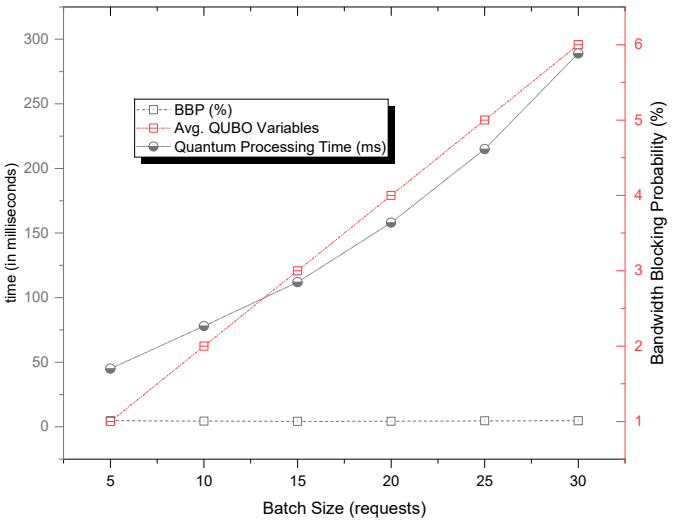


Figure 7. Effect of batch size on bandwidth blocking at a 200-Erlang load on the NSFNET topology.

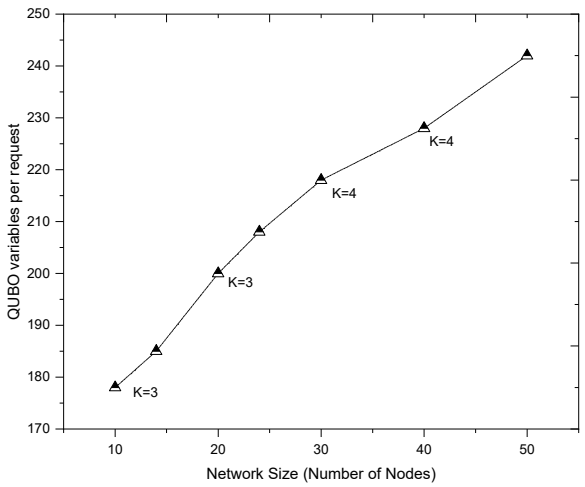


Figure 8. QUBO scaling with network size.

Additionally, note that the evaluation employs Poisson traffic with uniformly distributed bandwidth requests, which is standard for RSA benchmarking, but does not capture high-load fluctuations or non-uniform traffic patterns such as bursty arrivals. Future work will extend the analysis to self-similar traffic models and hotspot-demand scenarios to validate the framework's robustness under more challenging operational conditions.

Decision Space Analysis: Figure 9 presents a 3D scatter plot of the Pareto frontier at a 200-Erlang load on NSFNET. Each axis represents one objective: blocking probability (lower is better), spectral utilisation (lower is better), and fragmentation (higher is better). The plot shows that hybrid QA solutions (blue) form a well-distributed surface dominating MOCS solutions (orange), as most hybrid solutions exhibit lower blocking probability and

higher fragmentation. Thus, the 3D visualisation preserves the complete three-objective trade-off structure without projection distortion. As an example, we considered a batch with $|\ R\ | = 3$ requests. After feasibility pruning, suppose $N = 12$ binary variables. A particular non-dominated decision vector (corresponding to the Pareto point shown in Figure 9 with $\overline{f}_1 = 0.042$, $\overline{f}_2 = 0.31$, $\overline{f}_3 = 0.68$) is

$$x_{1,p_2,s_5} = 1,\ x_{2,p_1,s_8} = 1,\ x_{3,p_3,s_2} = 1,$$

all other $x_{r,p,s} = 0$, and $y_1 = y_2 = y_3 = 0$. This means request 1 is assigned to path p_2 starting at slot 5, request 2 to path p_1 starting at slot 8, and request 3 to path p_3 starting at slot 2, and no request is blocked. The decision space contains $2^{12} = 4096$ binary vectors; the Pareto set is the subset of those that are mutually non-dominated. Our hybrid quantum–classical method samples this subset via the QUBO Hamiltonian (12).

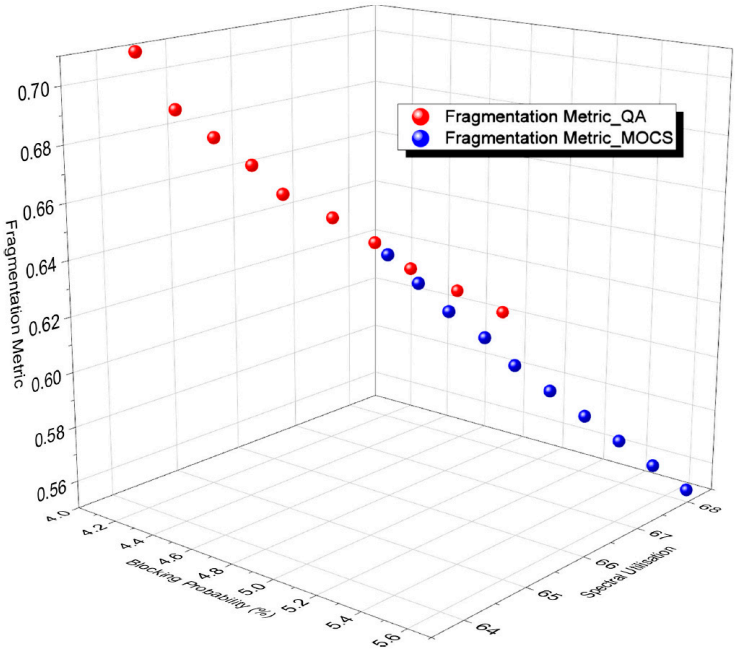


Figure 9. Pareto frontier in the normalised objective space $[0,1]^3$ for NSFNET at 200 Erlangs. Note: Each point corresponds to a non-dominated decision vector, $\boldsymbol{x}$, obtained by the hybrid QA framework. The three axes are $\overline{f}_1$ (blocking probability), $\overline{f}_2$ (spectral cost), and $1 - \overline{f}_3$ (fragmentation penalty, lower is better). Because all objectives are scaled to $[0,1]$, the weighted sum (6) treats each with comparable influence. This figure confirms that no single objective consistently dominates the others across the trade-off surface. A comprehensive comparison of all five metrics across algorithms is provided in Table 4 (classification) and Figures 2, 3 and 9 (objective values).

6. Conclusions

This paper introduces a hybrid quantum–classical framework for multi-objective RSA in EONs. By splitting the optimisation process into classical pre-processing, a quantum core (using either quantum annealing or the QAOA), and classical post-processing with Pareto management, we show significant practical benefits, even with NISQ hardware. Our main contributions include (1) a QUBO formulation that considers blocking probability, spectral efficiency, and fragmentation while fully enforcing RSA constraints; (2) a three-stage architecture that overcomes hardware limitations; (3) simulations demonstrating that the hybrid quantum algorithm achieves 14% lower blocking and 7.3% better fragmentation compared to classical heuristics; and (4) the identification of a crossover point around 220 variables where hybrid methods outperform classical meta-heuristics in both solution quality and

speed. Noise analysis indicates the QAOA retains 85–90% of its optimal performance despite current gate errors. These findings suggest network operators can use near-term quantum devices to improve resource utilisation without waiting for fault-tolerant quantum systems. While the evaluation is limited to NSFNET and UBN, which are standard benchmark topologies in optical networking, the framework's modular architecture and the demonstrated crossover point at 220 variables suggest that the approach can be extended to larger networks through hierarchical decomposition or by operating on subgraphs corresponding to regional network domains. Future work will explore scalability to larger topologies and integration with real-time network control planes.

Author Contributions: Conceptualisation, B.N. and B.P.; methodology, B.P.; software, B.N.; validation, B.N. and B.P.; formal analysis, B.P.; investigation, B.N. and B.P.; resources, B.N.; data curation, B.P.; writing—original draft preparation, B.P.; writing—review and editing, B.P. and B.N.; visualisation, B.P. and B.N.; supervision, B.N.; project administration, B.N.; funding acquisition, N/A or B.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Available on request.

Acknowledgments: The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BBP	Bandwidth blocking probability
COBYLA	Constrained Optimisation by Linear Approximation
DRL	Deep Reinforcement Learning
EON	Elastic Optical Network
FF	First Fit
ILP	Integer Linear Programming
IoT	Internet of Things
K SP FF	K-Shortest Path with First Fit
MILP	Mixed-Integer Linear Programming
MOCS	Multi-Objective Cuckoo Search
NISQ	Noisy Intermediate-Scale Quantum
NSFNET	National Science Foundation Network
OFDM	Orthogonal Frequency Division Multiplexing
QA	Quantum annealing
QAOA	Quantum Approximate Optimisation Algorithm
QUBO	Quadratic Unconstrained Binary Optimisation
RSA	Routing and spectrum assignment
RWA	Routing and Wavelength Assignment
SA	Simulated Annealing
UBN	United Backbone Network
VNE	Virtual Network Embedding
WDM	Wavelength Division Multiplexing

References

1. Cisco. Cisco Annual Internet Report (2018–2023). In *White Paper*; Cisco Systems, Inc.: San Jose, CA, USA, 2022; Available online: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html (accessed on 29 March 2026).
2. ITU-T. Requirements and Use Cases of Network 2030. In *Recommendation Y.3073*; International Telecommunication Union: Geneva, Switzerland, 2024; Available online: https://www.itu.int/rec/T-REC-Y.3073-202401-I (accessed on 30 March 2026).
3. Chen, X.; Li, B.; Proietti, R.; Lu, H.; Zhu, Z.; Yoo, S.J.B. DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks. *J. Light. Technol.* **2019**, *37*, 4155–4163. [CrossRef]
4. Chatterjee, B.C.; Ba, S.; Oki, E. Fragmentation problems and management approaches in elastic optical networks: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 183–210. [CrossRef]
5. Fernandes da Silva, P.H.; Kerivin, H.; Morvan, M.; Wagler, A. Routing and spectrum assignment in elastic optical networks integrating quality of transmission requirements. *Eur. J. Oper. Res.* **2026**, *330*, 268–278. [CrossRef]
6. Purnasha, S.; Mourya, N.; Srivastava, S.; Singh, A.K. Spectrum fragmentation mitigation in elastic optical networks: A chronologically structured review using a 7-point analytical framework. *J. Opt. Commun.* **2025**, *46*, 545–551. [CrossRef]
7. Bouchmal, O.; Cimoli, B.; Stabile, R.; Vegas Olmos, J.J.; Tafur Monroy, I. Quantum approximate optimization algorithm applied to multi-objective routing for large scale 6G networks. *Comput. Netw.* **2025**, *267*, 111345. [CrossRef]
8. Chatterjee, B.C.; Fadini, W.; Oki, E. A spectrum allocation scheme based on first–last-exact fit policy for elastic optical networks. *J. Netw. Comput. Appl.* **2016**, *68*, 164–172. [CrossRef]
9. Zheng, W.; Li, Y.; Wang, H.; Zhang, J.; Liu, Y.; Chen, X. Improved ILP models and heuristics for solving routing and resource allocation problems in optical networks. *J. Light. Technol.* **2025**, *43*, 3016–3033. [CrossRef]
10. Agarwal, R.; Bhatia, R. Multipath routing using genetic algorithm in elastic optical network. *J. Opt.* **2024**, *53*, 2316–2321. [CrossRef]
11. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. [CrossRef]
12. Kim, S.; Ahn, S.W.; Suh, I.S.; Dowling, A.W.; Lee, E.; Luo, T. Quantum annealing for combinatorial optimization: A benchmarking study. *npj Quantum Inf.* **2025**, *11*, 77. [CrossRef]
13. Blekos, K.; Brand, D.; Ceschini, A.; Chou, C.-H.; Li, R.-H.; Pandya, K.; Summer, A. A review on Quantum Approximate Optimization Algorithm and its variants. *Phys. Rep.* **2024**, *1068*, 1–66. [CrossRef]
14. Davies, E.; Banfield, D.; Weaver, B.; White, C.; Walker, N. Routing and wavelength assignment problems in optical networks—Comparing formulations for solution by quantum annealing. *J. Opt. Commun. Netw.* **2025**, *17*, B83–B91. [CrossRef]
15. Bouchmal, O.; Cimoli, B.; Stabile, R.; Vegas Olmos, J.J.; Hernandez-Chulde, C.; Martinez, R.; Casellas, R.; Tafur Monroy, I. Quantum computing approach for multi-objective routing and spectrum assignment optimization. *J. Opt. Commun. Netw.* **2025**, *17*, B15–B27. [CrossRef]
16. Villamayor-Paredes, M.M.R.; Maidana-Benítez, L.V.; Colbes, J.; Pinto-Roa, D.P. Routing, modulation level, and spectrum assignment in elastic optical networks. A route-permutation based genetic algorithms. *Opt. Switch. Netw.* **2023**, *47*, 100710. [CrossRef]
17. ProjectQ Contributors. ProjectQ: An Open Source Software Framework for Quantum Computing. GitHub. Available online: https://github.com/ProjectQ-Framework/ProjectQ (accessed on 30 March 2026).
18. Joshi, A.S.; Kulkarni, O.; Kakandikar, G.M.; Nandedkar, V.M. Cuckoo search optimization—A review. *Mater. Today Proc.* **2017**, *4*, 7262–7269. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.