

deadline aware hybrid optimization for ISP traffic engineering

conditional guarantees and a reproducible reference model

León Acosta
independent research

11 October 2026

Abstract

an optimizer can return a good routing configuration after the conditions that justified it have disappeared. we formulate a finite horizon decision problem in which an ISP chooses a traffic subset, solver and computation budget according to benefit remaining after computation and deployment. quantum annealing or a gate based optimizer may generate candidates; a classical acceptance layer remains responsible for capacity checks and the decision to apply a change. we prove conditional results for robust allocation and transition feasibility, composition of reserved subproblems, exact QUBO penalties, and selection of a positive expected improvement using independent validation data. we also derive a deadline break-even condition and give counterexamples to stronger claims. a dependency-free reference implementation checks small instances exhaustively. these are mathematical guarantees within stated models, not a proof of quantum advantage, production reliability, or a universally novel algorithm. no quantum hardware or real ISP measurements are reported

status. research preprint; not peer reviewed. proposed architecture plus elementary conditional analysis. the code contains mathematical checks and constructed examples, not an operational network controller

1 problem and contribution

IP traffic engineering allocates demands across network capacity; TCP separately provides transport reliability and congestion control. the routing objective can involve delay, loss, admitted traffic, cost or resilience [1]. a useful decision must include the work needed to compute, validate and install it. an optimal snapshot solution is not necessarily the best intervention over time

the proposed decision unit is a *job*: a subset of interacting demands, permitted paths, a solver, a resource budget, an expiry time, and an installation policy. the controller can keep the current baseline instead of submitting or deploying a job. quantum computation is an optional candidate generator; no theorem below depends on the device being quantum

the specific research hypothesis is that *jointly choosing the subset, solver and budget using benefit remaining after deployment* improves operational outcomes over a fixed solver and fixed optimization cadence. this paper makes that hypothesis testable and identifies sufficient conditions for accepting a candidate. it does not establish that the hypothesis is true on ISP workloads

the mathematical contribution is a composed specification: a route change must pass a resource certificate, a transition certificate, a freshness check, and an improvement test with an explicit comparator. the ingredients use standard arguments. their statement here is not a claim that penalty methods, robust bounds, concentration inequalities or hybrid optimization are new

superposition does not expose all routes for simultaneous classical inspection: a measurement produces a sample, not a list of all encoded solutions and their costs. a useful quantum algorithm must bias its samples toward useful candidates, and account for data preparation, repeated execution and decoding. this proposal concerns control-plane optimization of classical IP traffic; it does not route packets through a quantum network

2 relationship to existing work

Davies et al. formulate optical routing with latency and resilience constraints as QUBOs and demonstrate a hybrid annealing solver [2]. Harb et al. study resilient Layer 3 routing with a small QAOA demonstration [3]. neither result establishes large ISP speedup. Witt proposes quantum-compatible network reconfiguration in response to bursty load and evaluates it in a flow simulator [4]

Deutsche Telekom and Fujitsu’s published routing application describes QUBO objectives, segment routing, processing subsets of demands, residual capacities and deployment [5]. consequently, neither “quantum routing” nor “split it into smaller jobs” is an adequate novelty claim for this proposal. this is a technical comparison with a published document, not an assessment of patent scope

classical work already jointly optimizes traffic engineering and rerouting costs [6]. SWAN includes congestion-free network updates [7]. quantum warm starts [8] and predictive quantum solver selection [9] are also established. our narrower research question concerns their integration with a workload-dependent useful horizon and the measured cost of quantum access, validation and deployment. a complete systematic novelty review remains open; the sources reviewed do not justify a claim of priority

3 network, uncertainty and timing model

3.1 allocation model

let $G = (V, E)$ be a directed graph with capacities $C_e > 0$. let D be a finite set of aggregate demands. a classical path generator supplies a nonempty finite set P_i of simple, policy-compliant paths for demand i , including its current path. write $a_{ipe} \in \{0, 1\}$ for path incidence. path generation and policy validity are assumed, not inferred from a binary incidence vector for the selected subset $S \subseteq D$, $x_{ip} = 1$ selects path p for i . other demands remain fixed, with reserved upper load F_e . let $\bar{d}_i$ upper-bound selected demand throughout the whole decision horizon. robust final feasibility is

$$\sum_{p \in P_i} x_{ip} = 1 \quad (i \in S), \quad (1)$$

$$F_e + \sum_{i \in S} \bar{d}_i \sum_{p \in P_i} a_{ipe} x_{ip} \leq C_e \quad (e \in E). \quad (2)$$

because incidences are nonnegative, these inequalities imply final capacity feasibility for every

$0 \leq d_i(t) \leq \bar{d}_i$. the box model permits all upper bounds simultaneously; correlations can make it conservative

the paper studies indivisible aggregate path choices. allowing continuous splitting can yield a tractable linear formulation for some objectives. such a formulation is a required classical baseline whenever the operator can implement it; restrictions must come from the deployment problem, not a desire to favor quantum methods

3.2 freshness and envelopes

one sufficient way to obtain a demand bound is to assume measurement error at most η_i and demand drift at most ρ_i per unit time. for an observation $\hat{d}_i$ at time $t_i \leq t_0$, a horizon of length $H > 0$ gives

$$\bar{d}_i = \hat{d}_i + \eta_i + \rho_i(t_0 + H - t_i). \quad (3)$$

the implication follows from the triangle inequality if the error and drift assumptions hold. it does not establish those assumptions. a measured percentile is not a hard envelope. unbounded bursts, failures or unreserved external traffic invalidate the capacity guarantee

each job carries the routing and reservation version used to build it. an atomic version check and reservation commit must prevent another update from changing those conditions between validation and installation. elapsed time alone does not detect concurrent changes. any unexpected topology or reservation change cancels the job and requires a new model. changing the baseline, horizon or scenario distribution also requires revisiting the improvement certificate; its value is specific to those conditions

3.3 operational objective

fix a horizon $[0, H]$ relative to t_0 and a classical baseline policy π_0 . this can be a fixed allocation or a specified dynamic controller. use a bounded operational loss $\ell_\pi(t, \omega) \in [0, 1]$, where ω is a future traffic and failure scenario. the chosen metric and its normalization must be declared; bounded loss is not a claim that raw latency is bounded

candidate policy π_j includes subset selection, solver behavior, timeouts, installation, fresh-state checks and fallback. its sampled average improvement and net expected value are

$$Z_j(\omega) = \frac{1}{H} \int_0^H [\ell_{\pi_0}(t, \omega) - \ell_{\pi_j}(t, \omega)] dt \in [-1, 1], \quad (4)$$

$$J_j = \mathbb{E}[Z_j] - c_j, \quad J_0 = 0. \quad (5)$$

$c_j \geq 0$ is a declared normalized computation/control cost or a deterministic upper bound on that cost. transition-induced service loss belongs inside ℓ , not again in c_j . money, delay and loss cannot be subtracted without an explicit conversion. unsuccessful jobs still incur computation costs. rejected route changes can preserve service behavior while making net value negative

the formal decision problem is to maximize J_j over a finite menu of policies. exact J_j is generally unavailable. the proposal uses conservative estimates; it has no general optimality or competitive-ratio guarantee for online scheduling

4 a deadline condition

Proposition 1 (benefit remaining after readiness). *suppose a candidate becomes ready at $\tau \geq 0$, behaves identically to the baseline before τ , has constant loss reduction $g \geq 0$ afterwards until H , and incurs total cost $K \geq 0$ in loss-time units. its unnormalized net benefit is*

$$B = g[H - \tau]_+ - K. \quad (6)$$

for $g > 0$, $B > 0$ if and only if $\tau < H - K/g$. no $\tau \geq H$ can yield positive benefit. if post-readiness reduction is only lower-bounded by g and additional transient loss is bounded by A , then $g[H - \tau]_+ - K - A$ is a lower bound

Proof. the loss difference integrates to zero before τ and to $g(H - \tau)$ afterwards when $\tau < H$, otherwise zero. subtracting K proves the equality. rearranging the strict inequality proves the deadline condition. integration preserves a pointwise lower bound, and subtracting an upper bound on transient loss yields the final statement $\square$

this proposition is a screening rule under its assumptions, not a forecast. the full policy objective in (4) covers variable traffic, random job completion and installation effects. a promised solver duration cannot replace a measured or justified runtime distribution

5 capacity and transition certificates

5.1 asynchronous fluid transition

let a_{ie}^- and a_{ie}^+ be old and new path incidences. during a transition assume a coherent fluid split $\alpha_i(t) \in [0, 1]$ for each demand, so that its modeled load on edge e is

$$d_i(t) \left[(1 - \alpha_i(t))a_{ie}^- + \alpha_i(t)a_{ie}^+ \right]. \quad (7)$$

the same $\alpha_i(t)$ applies along the complete path; different demands may transition independently. define

$$U_e = F_e + \sum_{i \in S} \bar{d}_i \max\{a_{ie}^-, a_{ie}^+\}. \quad (8)$$

Theorem 2 (robust transition feasibility in the fluid model). *under (7) and the demand box, U_e is the exact maximum possible modeled load on each edge. therefore $U_e \leq C_e$ for every e is necessary and sufficient for capacity feasibility for every allowed fluid split and demand vector*

Proof. the convex combination in (7) is at most the larger incidence. multiplying by $d_i \leq \bar{d}_i$ and summing gives the upper bound. for any fixed edge, choose $d_i = \bar{d}_i$ and choose α_i at an endpoint that attains the larger incidence. the box and independent split assumptions allow these choices simultaneously for that edge. thus its bound is attained. feasibility for all edges and all choices is equivalent to each edge's maximum respecting its capacity $\square$

scope. this is not a packet-level theorem. propagation-delay differences, packets in flight, queue bursts and inconsistent forwarding updates can violate the common- α representation. deployment requires either enforcement of this model or independently justified per-edge arrival envelopes. prevalidated simple paths do not by themselves make an arbitrary switch-by-switch update loop-free. consistent forwarding, sufficient drain intervals and reservation enforcement are separate obligations

counterexample. two unit demands occupy separate unit-capacity links A and B and exchange paths. both endpoint allocations are feasible, but if one demand moves first, one link carries two units. (8) gives $U_A = U_B = 2$ and rejects the simultaneous unscheduled exchange. a spare unit-capacity path C allows the completed stages $A \rightarrow C$, $B \rightarrow A$, $C \rightarrow B$; each stage passes the fluid envelope test. this demonstrates why final feasibility is insufficient

5.2 composition of subproblems

Theorem 3 (reserved capacity composition). *partition selected demands into disjoint groups $S_1, \dots, S_m$. assign group budgets $q_{ge} \geq 0$ with $F_e + \sum_g q_{ge} \leq C_e$. if each group's actual load is at most q_{ge} throughout its accepted transition, then concurrently executing all groups respects capacities. a sufficient condition in the fluid model is to reserve each group's envelope from (8), excluding F_e , within q_{ge}*

Proof. disjoint membership avoids counting the same traffic as two independent jobs. for every edge and instant, total load is at most $F_e + \sum_g q_{ge} \leq C_e$. Theorem 2 supplies a sufficient group bound $\square$

independently testing each job against the *same* spare capacity is not reservation: two jobs can each consume that capacity and exceed it together. fixed group budgets ensure feasibility but can reduce objective quality. no global optimality follows from decomposition

6 a quantum-compatible candidate encoding

the classical controller may use an integer solver or a QUBO candidate generator. for one selected subset, assume $\bar{d}_i$, C_e and F_e are nonnegative integers in common units, with residual $R_e = C_e - F_e \geq 0$. conservative rounding uses upward demand bounds and downward capacities. route and resource feasibility must also be checked in original units

let $f(x)$ be a quadratic surrogate objective, such as weighted link-load squares plus path costs and change penalties. it is a candidate-ranking objective, not generally equal to the scenario objective J . suppose known bounds satisfy $f_{\min} \leq f(x) \leq f_{\max}$ on *all* binary route assignments, including infeasible assignments. encode slack as

$$s_e = \sum_{k=0}^{b_e-1} 2^k y_{ek}, \quad b_e = \lceil \log_2(R_e + 1) \rceil, \quad y_{ek} \in \{0, 1\}. \quad (9)$$

when $R_e = 0$, $b_e = 0$ and $s_e = 0$. the QUBO is

$$\begin{aligned} E_P(x, y) = f(x) + P \sum_{i \in S} \left(\sum_{p \in P_i} x_{ip} - 1 \right)^2 \\ + P \sum_{e \in E} \left(\sum_{i \in S, p \in P_i} \bar{d}_i a_{ipe} x_{ip} + s_e - R_e \right)^2. \end{aligned} \quad (10)$$

all squared terms contain linear expressions, so the polynomial is quadratic on binary variables

Theorem 4 (sufficient exact penalty). *assume the allocation constraints have a feasible solution and arithmetic is exact. if $P > f_{\max} - f_{\min}$, every global minimizer of (10) decodes to a feasible allocation minimizing f over feasible allocations*

Proof. every feasible allocation has integer slack $s_e \in [0, R_e]$, representable by (9); it has zero penalty and energy at most $f_{\max}$. every assignment with nonzero penalty has at least one nonzero integer residual, hence total squared residual at least one. its energy is at least $f_{\min} + P > f_{\max}$. it cannot be globally minimal. zero-penalty assignments are exactly the constraints with representable nonnegative slack, and minimizing their energy minimizes f $\square$

the strict inequality matters: for a single required variable with $f(x) = 2x$, penalty $P = 2$ can tie an invalid assignment with the valid one. if the problem is infeasible, no penalty manufactures a feasible solution. finite precision and device noise are outside this theorem. large penalties can harm usable coefficient resolution; therefore a mathematical bound is not a hardware performance recommendation

6.1 resources and solver contract

the encoding requires

$$N = \sum_{i \in S} |P_i| + \sum_{e \in E} \lceil \log_2(R_e + 1) \rceil \quad (11)$$

binary variables before additional constraints. for 100 demands with four candidate paths and 50 links with $R_e = 100$, $N = 400 + 350 = 750$. this is an encoding count, not a minimum or a physical qubit estimate. QAOA would directly use N data qubits for this representation before implementation overhead. annealing embedding may use chains of physical qubits per variable [10]. fault tolerant circuit resources further depend on the circuit and error correction

the solver returns candidates and measured runtime, not a trusted feasibility or optimality assertion. exact integer validation rejects invalid samples. even a feasible global minimum of f still requires transition and operational-benefit checks. a heuristic or noisy sampler is not guaranteed to find the ground state; Theorem 4 gives no sample complexity or quantum advantage

the reference code uses a linear $f(x) = \sum c_{ip}x_{ip}$. its exact all-binary range is $\sum |c_{ip}|$, including negative coefficients. it chooses $P = 1 + \sum |c_{ip}|$. the more general quadratic theorem is not a claim that the prototype implements every possible surrogate

6.2 reliability and scaling

there is no defensible single physical qubit count for reliable deployment without a compiled algorithm, target error, connectivity, runtime and workload. for example, if a fault tolerant computation has G_{loc} logical fault locations, each with error probability at most p_L , a union bound gives

$$\Pr(\text{one or more logical faults}) \leq G_{\text{loc}}p_L. \quad (12)$$

achieving a chosen fault budget ε_{hw} therefore has the sufficient condition $p_L \leq \varepsilon_{\text{hw}}/G_{\text{loc}}$. this does not specify the physical qubits needed to reach that error rate, nor ensure that a fault-free heuristic returns a useful answer. solver failure, model error and hardware error are distinct

scaling the architecture means aggregating traffic into operational demands, generating a few permitted paths classically, scheduling interacting subsets, and allocating explicit link budgets to concurrent jobs. each quantum job then fits a bounded encoding while a classical coordinator handles global resources. Theorem 3 supplies conditional capacity safety, but neither near-linear throughput scaling nor preservation of global solution quality. finer traffic units, more candidate paths and finer capacity precision all raise resource costs; large, tightly coupled bottlenecks can defeat decomposition

7 a statistically valid acceptance rule

freeze a menu of $M \geq 1$ complete candidate policies independently of validation data. conditional on the training data and current modeled state, let $\omega_1, \dots, \omega_n$ be independent scenarios from the same distribution that defines deployment value. evaluate all policies on these held-out scenarios; they may share scenarios. let $\hat{\mu}_j = n^{-1} \sum_k Z_j(\omega_k)$, and for $0 < \delta < 1$ define

$$\epsilon = \sqrt{\frac{2 \log(2M/\delta)}{n}}, \quad L_j = \hat{\mu}_j - \epsilon - c_j. \quad (13)$$

choose a policy maximizing L_j only if its bound is strictly positive; otherwise choose baseline policy 0

Theorem 5 (finite-menu expected-improvement certificate). *under the independence, bounded-loss and distribution assumptions above, with probability at least $1 - \delta$ over validation data, every candidate satisfies $J_j \geq L_j$. on this event the selected policy has positive expected net value if one is accepted, and value zero when the baseline is selected*

Proof. for a fixed policy, Hoeffding's inequality on variables in $[-1, 1]$ gives

$$\Pr(|\hat{\mu}_j - \mathbb{E}Z_j| > \epsilon) \leq 2 \exp(-n\epsilon^2/2) = \delta/M.$$

a union bound over the M policies gives simultaneous coverage with probability at least $1 - \delta$; independence between policies is unnecessary. on that event, subtracting c_j yields $J_j \geq L_j$ for all j , including a policy selected after seeing the bounds. a positive bound implies positive expected value. the baseline has defined value zero $\square$

this is a direct application of a classical concentration inequality [11], not a new concentration result. the probability concerns validity of an *expected-value claim*; it is not a $1 - \delta$ guarantee that each realized episode improves. the policy must already include timeouts, compute costs and release gates. adding an unmodeled acceptance condition afterwards changes the policy and invalidates the argument

Corollary 6 (repeated decisions with valid fresh validation). *if decision r satisfies the theorem conditionally on its past, using failure budget δ_r with $\sum_r \delta_r \leq \delta_{\text{total}}$, then all its expected-value certificates are valid simultaneously with probability at least $1 - \delta_{\text{total}}$*

Proof. conditional failure probabilities are at most δ_r , so their unconditional probabilities are also at most δ_r . the union bound, including its countable limit if needed, gives the result $\square$

reusing a trace to train policies, tune budgets and certify them does not satisfy these assumptions. adjacent samples in a traffic time series are not automatically independent episodes. distribution shift or inaccurate simulation can invalidate the certificate entirely. historical replay therefore yields a benchmark result, not an operational guarantee, unless the sampling and deployment assumptions are independently justified. abstention may be common: the bound can be too wide for small gains

8 controller procedure and scope of the guarantee

1. define the comparator, loss, horizon, envelopes and snapshot version; check that the baseline remains admissible
2. construct interacting demand groups and a finite menu of solver/budget policies. reserve capacity if jobs will run concurrently
3. use training data to choose candidate generation and timing rules. validate the frozen policies on independent scenarios where justified
4. execute only a policy with positive certified or otherwise explicitly labeled estimated value. a proof-backed certificate and an empirical estimate are distinct modes
5. when a candidate returns, reject invalid, expired or version-mismatched results. enforce the predeclared transition procedure and reserved envelopes before committing
6. record the complete computation, validation and deployment time, all rejected jobs, and the resulting service loss

the reference implementation provides predicates for these checks; it does not implement forecasting, atomic reservations, device configuration, online solver scheduling, or the generation of representative future scenarios

Proposition 7 (conditional composition). *suppose an accepted policy has a valid certificate from Theorem 5; its predeclared execution enforces Theorem 2 at every completed stage, Theorem 3 for concurrent groups, and version-consistent reservations. then modeled capacity constraints hold throughout its execution. with the certificate’s confidence, its expected normalized net benefit is positive*

Proof. each transition stage respects its reserved capacities by Theorem 2. stages start only from the completed predecessor. concurrent disjoint groups compose by Theorem 3, and version-consistent reservations preserve the premises. separately, Theorem 5 supplies the expected-value claim for the complete policy. neither claim implies the other $\square$

an expired or failed job may safely leave a *still feasible* baseline unchanged. a link failure can make that baseline infeasible. existing emergency recovery must handle that case; “fallback” is not a proof of availability

Proposition 8 (no strict improvement guarantee for arbitrary futures). *without restrictions on future losses, a policy cannot guarantee strict benefit over staying with a baseline in every possible future*

Proof. consider two paths with identical observations. one possible future assigns zero loss to the baseline path and positive loss to the alternative for the whole horizon. staying obtains zero improvement; moving can only worsen performance in that future, before any nonnegative computation cost. hence neither staying nor a policy with positive probability of moving guarantees strict improvement in all futures $\square$

the assumption of a predictable useful horizon or a defensible scenario distribution is substantive, not a detail that more qubits remove

9 reproducible checks and constructed examples

all reported numbers are generated by `python3 scripts/reproduce.py`. the tests use Python’s standard library and exact integer arithmetic for QUBOs; no network, QPU, paid

API or external dataset is needed. the exhaustive checks are finite regression evidence, not a machine-checked proof of the theorems

check	observed constructed result
QUBO example	8 variables, 256 assignments enumerated; unique minimum decodes to a feasible allocation
asynchronous swap	both endpoints feasible; direct envelope $(2, 2, 0)$ exceeds capacities $(1, 1, 1)$; three stages through a spare path pass
deadline ranking	$H = 30$: $(g, \tau, K) = (0.1, 2, 0.2)$ yields 2.6; $(0.3, 25, 0.2)$ yields 1.3 loss-time units
certificate arithmetic	$M = 2, n = 1000, \delta = 0.05$, constant constructed gains 0.4, 0.1 and cost 0.05 each; first lower bound 0.2564

the constant validation examples exercise arithmetic and selection, not predictive accuracy. an additional test computes the exact false-acceptance probability for a specified iid two-point distribution with negative mean and compares it with the stated bound

the test suite checks polynomial expansion against the squared-residual definition on every assignment for 64 seeded feasible small instances, including negative costs; it compares QUBO minima with independent feasible route enumeration. it also checks strict-penalty failure, infeasible inputs, zero slack width, transition envelopes at all binary corners, stale versions, expiry and abstention. the tiny instances are intentionally small enough for exact checking, not representative of a difficult quantum workload

10 evaluation needed before a performance claim

compare a strong classical optimizer; the proposed selection and timing method using classical solvers; and the identical method with a quantum backend. also compare a classical solver on the same QUBO and candidate sets. this isolates improvements due to representation, controller logic and hardware. include linear optimization when traffic can split, and integer optimization when actual control granularity requires discrete decisions

use separate training and test periods, controlled traffic changes, failures and burst episodes. measure accumulated service loss over time, rejected/expired work, route changes and full resource cost. report hardware access and queue delay, preprocessing, embedding or circuit compilation, sampling, classical parameter tuning, validation and installation. quantum circuit simulation is classical execution and must be labeled accordingly

an offline long-budget solution can supply a reference value or a bound; it is not a deployable comparator with free future knowledge. no benchmark should grant only the quantum method a better path generator, warm start or preprocessing budget. publish unsuccessful cases and compute an uncertainty interval across independent episodes where that independence is defensible

11 limitations and next research questions

the main limitations are a conservative box uncertainty model, an idealized fluid transition model, a finite menu, potentially loose statistical bounds and unproven workload novelty. the

framework does not guarantee low latency, fairness or TCP stability unless those properties are modeled and validated separately. it does not prove convergence of QAOA, practical qubit sufficiency, a sample-complexity improvement, or quantum speedup. error correction would address hardware faults, not these missing optimization results

the most useful next steps are to develop a topology-aware grouping method with measured loss of solution quality; construct tighter transition envelopes valid for packet traffic; estimate useful horizons without leaking test data; and locate workloads where computation is actually the limiting cost. the project should retain its classical value even if the quantum backend offers no benefit. a claimed quantum contribution requires additional evidence beyond the conditional correctness developed here

authorship and assistance

the research direction was proposed by León Acosta. this draft, mathematical exposition and reference code were developed with assistance from OpenAI ChatGPT/Codex. the work has not received independent expert review or proof-assistant verification. publication here does not imply experimental validation

References

- [1] IETF. *RFC 9522: Overview and Principles of Internet Traffic Engineering*. 2024.
<https://www.rfc-editor.org/rfc/rfc9522.html>
- [2] E. Davies et al. *Optical Routing with Binary Optimisation and Quantum Annealing*. 2024.
<https://arxiv.org/abs/2402.07600>
- [3] M. Harb et al. *Quantum-Based Resilient Routing in Networks: Minimizing Latency Under Dual-Link Failures*. 2026. <https://arxiv.org/abs/2602.04495>
- [4] A. Witt. *Queue-aware Network Control Algorithm with a High Quantum Computing Readiness—Evaluated in Discrete-time Flow Simulator for Fat-Pipe Networks*. 2024.
<https://arxiv.org/abs/2404.04080>
- [5] M. Geitz et al.; Deutsche Telekom AG and Fujitsu Technology Solutions GmbH. *Method for optimizing a routing in a communications network*. Publication WO2023041643A1, 2023.
<https://patents.google.com/patent/WO2023041643A1/en>
- [6] J. Zheng, Y. Xu, L. Wang, H. Dai and G. Chen. *Online Joint Optimization on Traffic Engineering and Network Update in Software-defined WANs*. IEEE INFOCOM, 2021.
<https://doi.org/10.1109/INFOCOM42981.2021.9488837>
- [7] C.-Y. Hong et al. *Achieving High Utilization with Software-Driven WAN*. ACM SIGCOMM, 2013.
<https://www.microsoft.com/en-us/research/publication/achieving-high-utilization-with-software-driven-wan/>
- [8] D. J. Egger, J. Mareček and S. Woerner. *Warm-starting quantum optimization*. Quantum 5:479, 2021. <https://doi.org/10.22331/q-2021-06-17-479>
- [9] D. Volpe, N. Quetschlich, M. Graziano, G. Turvani and R. Wille. *A Predictive Approach for Selecting the Best Quantum Solver for an Optimization Problem*. 2024.
<https://arxiv.org/abs/2408.03613>
- [10] D-Wave. *Minor Embedding*. documentation accessed 11 October 2026.
https://docs.dwavequantum.com/en/latest/quantum_research/embedding_intro.html
- [11] W. Hoeffding. *Probability Inequalities for Sums of Bounded Random Variables*. Journal of the American Statistical Association 58(301):13–30, 1963.
<https://doi.org/10.1080/01621459.1963.10500830>