WHITEPAPER · QUANTUM COMPUTING

Quantum Optimization for Logistics & Finance

A Near-Term Advantage Assessment

An honest, quantitative appraisal of where quantum and quantum-inspired methods help with hard combinatorial optimization today — what the NISQ era can and cannot deliver, and a realistic horizon for fault-tolerant advantage.

AUTHORS
Root Digit Quantum Lab

PUBLISHED
2026

VERSION
1.0

CLASSIFICATION
Public

rootdigit.com

WHITEPAPER

Abstract & Contents

ABSTRACT

Combinatorial optimization sits at the financial heart of logistics and finance: vehicle routing, crew and shift scheduling, network design, and portfolio construction are all NP-hard problems whose solution quality maps directly to cost and risk. Quantum computing is frequently marketed as the answer. This whitepaper takes a deliberately candid position. We describe the gate-model and quantum-annealing paradigms, the algorithms most often proposed for optimization — QAOA, quantum annealing, and VQE — and the quantum-*inspired* classical methods that run on conventional hardware today. We then assess, without hype, where advantage is real and where it is not. Our conclusion is that present-day Noisy Intermediate-Scale Quantum (NISQ) hardware does not yet deliver a robust, scaling advantage over state-of-the-art classical solvers for industrial optimization at useful sizes; the demonstrable near-term value lies in **hybrid and quantum-inspired** methods. Fault-tolerant quantum advantage for these problem classes is a multi-year horizon, not a current capability. All figures herein are modeled and illustrative, intended to convey structure and rank-order rather than vendor benchmarks.

KEY FINDINGS AT A GLANCE

- NISQ devices are limited by qubit count, gate noise, and sparse connectivity; for industrial optimization sizes they do not yet beat strong classical solvers.
- Quantum-inspired methods (simulated/digital annealing, tensor-network and GPU solvers) deliver measurable value *now* on classical hardware.
- Mapping a real problem to a QUBO/Ising form is the hard part; constraint penalties and minor-embedding inflate qubit requirements by an order of magnitude or more.
- A realistic horizon for fault-tolerant quantum advantage on routing, scheduling and portfolio problems is several years out; invest in pilots and skills, not production reliance.

Contents

SECTION ONE

1. Introduction: The Optimization Problems That Matter

Behind almost every logistics and finance operation sits a combinatorial optimization problem — a search over an astronomically large set of discrete choices for the one configuration that minimises cost or risk. These problems are not academic curiosities; they are where margin is won and lost.

Consider three families that recur across industries. The **vehicle routing problem (VRP)** and its variants ask how a fleet should visit a set of locations subject to capacity, time-window, and driver-hour constraints — a generalisation of the travelling salesman problem that governs last-mile delivery, field service, and waste collection. **Scheduling** problems assign people, machines, or aircraft to tasks and time slots under hard rule constraints; crew rostering, job-shop scheduling, and shift planning all live here. In finance, **portfolio optimization** selects asset weights to balance expected return against risk, and in its cardinality-constrained or integer-lot forms becomes genuinely combinatorial rather than the smooth convex problem of the textbook.

What unites them is computational hardness. Each is, in its general form, NP-hard: the number of candidate solutions grows combinatorially with problem size, and no known algorithm solves all instances to proven optimality in polynomial time. A routing problem with 50 stops has more possible tours than there are atoms in the observable universe. This does not mean such problems are unsolvable in practice — decades of operations-research engineering have produced remarkably effective solvers — but it does mean that solution quality is always a trade-off against compute time, and that even small percentage improvements translate into large absolute savings at enterprise scale. A logistics operator that shaves two percent off total distance driven across a national fleet, or an asset manager that trims tracking error while respecting lot constraints, is capturing value that compounds every single day.

It is worth being precise about why these problems resist brute force. The difficulty is not merely the size of the search space — modern solvers never enumerate it — but its *structure*. Combinatorial landscapes are riddled with local optima: configurations that cannot be improved by any small change yet fall well short of the global best. Constraints carve the feasible region into awkward, disconnected shapes. Integer decisions create cliffs where a tiny change in inputs forces a wholesale change in the optimal answer. These properties defeat the smooth, gradient-following methods that work so well in continuous optimization, which is why the field relies on a different toolkit: exact branch-and-bound search, cutting planes, and a rich family of metaheuristics that escape local optima by accepting occasional backward steps.

That hardness is precisely why quantum computing attracts attention. The promise, stated loosely, is that quantum hardware can explore exponentially large solution spaces in ways classical hardware cannot, and so find better solutions faster. The reality is more nuanced, and the gap between promise and present capability is the subject of this whitepaper. Optimization is, somewhat awkwardly, the application area where quantum computing is most heavily marketed and least clearly proven. There are good theoretical reasons to expect quantum methods to help eventually; there are equally good practical reasons to be sceptical of claims of advantage today.

Our objective is to equip the operations, technology, and investment leaders who must decide whether — and how much — to invest in quantum optimization with a clear, honest framing. We will explain how the hardware works, how problems are mapped onto it, what the algorithms actually do, and where the genuine near-term value lies. Throughout, we distinguish carefully between three things that are frequently conflated: true quantum hardware, hybrid quantum-classical methods, and quantum-*inspired* classical algorithms that borrow the mathematics of quantum optimization but run on ordinary CPUs and GPUs. The distinction matters enormously for what an enterprise can deploy and rely upon in 2026.

2. A Quantum Computing Primer

To assess quantum optimization honestly, one must first understand what the hardware is — and, just as importantly, what it is not. The vocabulary is unfamiliar, but the intuition is accessible.

Classical computers store information in bits, each definitively 0 or 1. A **qubit** can exist in a *superposition* of both states until measured; a register of n qubits can represent a weighted combination of all 2^n classical states at once. Two further phenomena give quantum computing its power. **Entanglement** correlates qubits so that the state of one cannot be described independently of another, allowing computations to exploit structure across the whole register. **Interference** lets a well-designed algorithm amplify the probability amplitudes of correct answers and cancel those of wrong ones. A quantum algorithm is, in essence, a careful choreography of superposition, entanglement, and interference so that a final measurement is likely to return a useful result.

Two hardware paradigms matter for optimization. The **gate (circuit) model** is the universal form: computation proceeds as a sequence of quantum logic gates applied to qubits, analogous to logic gates in a classical processor. It is general-purpose and is the basis of famous algorithms such as Shor's and Grover's. **Quantum annealing** is a specialised paradigm that does not run arbitrary circuits; instead it physically evolves a system of qubits toward the lowest-energy configuration of a programmable energy landscape, which can be arranged to encode an optimization problem. Annealers (notably from D-Wave) reach thousands of qubits but solve only a narrow class of problems, whereas gate-model machines are general but have far fewer usable qubits.

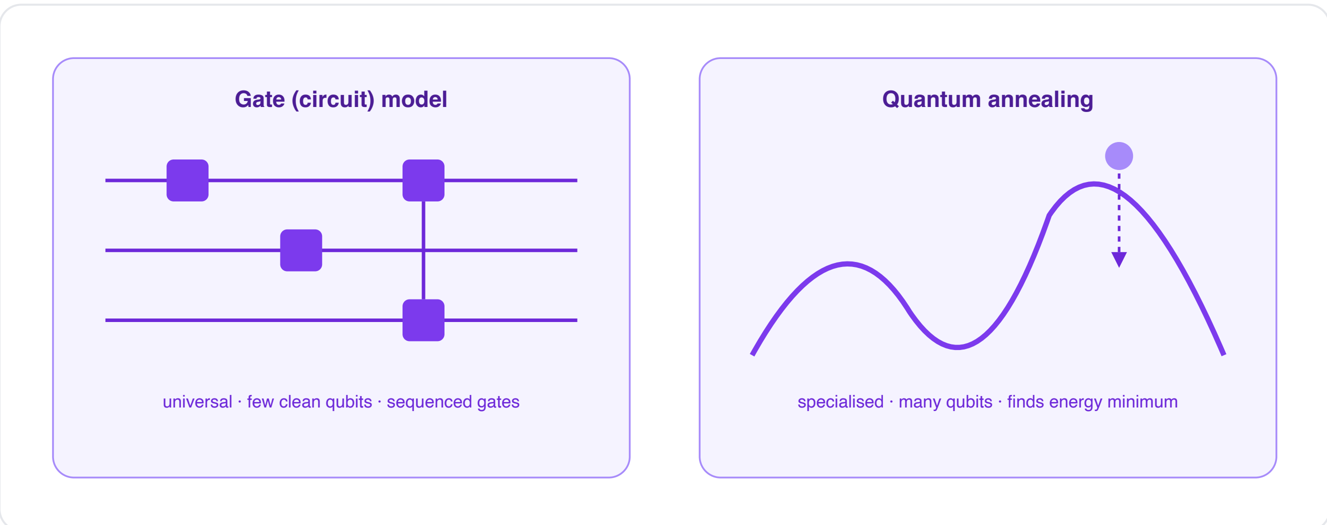

Figure 2.1 — Two hardware paradigms. The gate model runs arbitrary circuits on few high-quality qubits; the annealer relaxes many qubits toward the minimum of a programmable energy landscape.

The defining reality of the present is **noise**. Qubits are exquisitely fragile: stray interaction with the environment — heat, vibration, electromagnetic fields — destroys their quantum state in microseconds, a process called *decoherence*. Every gate operation has a non-trivial error rate, typically a fraction of a percent on the best hardware, and those errors compound as circuits grow deeper. The result is that today's machines can run only shallow circuits before the signal is swamped by error. This is the essence of the term **NISQ** — Noisy Intermediate-Scale Quantum — coined to describe the current era: devices with tens to a few hundred qubits, too noisy to run the large, deep, error-corrected algorithms that would deliver textbook quantum advantage.

The eventual answer is **fault-tolerant quantum computing**, in which many noisy physical qubits are combined into a single robust *logical* qubit using quantum error correction. The overhead is severe — current estimates require hundreds to thousands of physical qubits per logical qubit — which is why fault-tolerant machines at useful scale remain a multi-year engineering goal rather than a present reality. Understanding this distinction between noisy NISQ hardware and the fault-tolerant future is the single most important context for everything that follows.

SECTION THREE

3. Optimization Approaches

Several algorithmic families are proposed for optimization, spanning true quantum hardware, hybrid quantum-classical loops, and purely classical methods that borrow quantum mathematics. Knowing which is which is essential to a sober assessment.

On gate-model hardware, the leading candidate is the **Quantum Approximate Optimization Algorithm (QAOA)**. QAOA is a *hybrid* method: a parameterised quantum circuit prepares a candidate distribution over solutions, a classical optimiser tunes the circuit parameters to lower the measured cost, and the loop repeats. It is designed for the NISQ era because the circuits are shallow. The closely related **Variational Quantum Eigensolver (VQE)** uses the same hybrid pattern to find the lowest-energy state of a system; though best known for chemistry, its machinery applies to optimization problems cast in energy-minimisation form.

```
# QAOA hybrid loop (conceptual)
def qaoa(cost_hamiltonian, p_layers):
    params = init_angles(p_layers)
    while not converged(params):
        circuit  = build_circuit(cost_hamiltonian, params)  # quantum
        samples  = run_on_qpu(circuit, shots=4096)          # noisy
        expval   = estimate_cost(samples, cost_hamiltonian)
        params   = classical_optimizer.step(expval)         # classical
    return best_bitstring(samples)
```

Listing 3.1 — QAOA is a hybrid loop: the quantum processor proposes, a classical optimiser steers. Most of the intelligence lives in the classical half.

Quantum annealing takes a different route. Rather than building circuits, it encodes the optimization problem as an energy landscape and lets the hardware physically settle toward a low-energy state. It maps naturally to problems already expressed in Ising or QUBO form (Section 4) and scales to thousands of qubits, but its limited connectivity and analogue noise mean it does not reliably reach global optima for hard instances, and independent benchmarks against strong classical solvers have been mixed at best.

The most practically important category for enterprises today is **quantum-inspired classical optimization**. These algorithms take the mathematics that motivates quantum methods — energy minimisation over Ising models, for instance — and implement it on conventional CPUs, GPUs, FPGAs, or purpose-built digital silicon. **Simulated annealing**, a decades-old technique, mimics the physical annealing process classically. **Digital annealers** (such as Fujitsu's) and GPU-accelerated Ising solvers run the same QUBO formulation at very large scale on classical silicon. **Tensor-network methods** exploit the structure of quantum states to approximate solutions efficiently. Crucially, these run today, on hardware enterprises already own or can rent, and they often match or beat the quantum hardware on the very benchmarks quantum vendors cite — a point we return to candidly in Section 5.

REALITY CHECK

When a quantum vendor reports a benchmark, ask one question first: *what classical baseline did they compare against?* Many headline "quantum advantage" results compare new quantum hardware to weak or default classical solvers. Against a well-tuned classical or quantum-inspired solver, the advantage frequently shrinks or disappears. The honest framing for 2026 is that quantum-inspired methods are the deployable technology and true quantum hardware is a research bet.

The taxonomy, then, is three layers deep: pure quantum (annealing, and gate-model algorithms like QAOA in their quantum part), hybrid (QAOA and VQE as full loops, plus emerging hybrid solvers), and quantum-inspired classical. An enterprise optimization strategy in 2026 should be built on the bottom two layers, with the top layer treated as an experiment to learn from, not a system to depend on.

SECTION FOUR

4. Mapping Problems to Hardware

Before any quantum or quantum-inspired solver can touch a routing or portfolio problem, that problem must be translated into the one mathematical form the hardware understands. This translation is where most of the practical difficulty — and most of the hidden cost — lives.

The lingua franca is the **QUBO** (Quadratic Unconstrained Binary Optimization) and its physics-equivalent, the **Ising model**. A QUBO expresses the objective as a quadratic function of binary variables $x_i \in \{0,1\}$; the Ising model uses spin variables $s_i \in \{-1,+1\}$, and the two are related by the simple substitution $s = 2x - 1$. Both annealers and QAOA are, at bottom, machines for finding the assignment of these variables that minimises the energy of the model. The art of quantum optimization is therefore largely the art of writing a good QUBO.

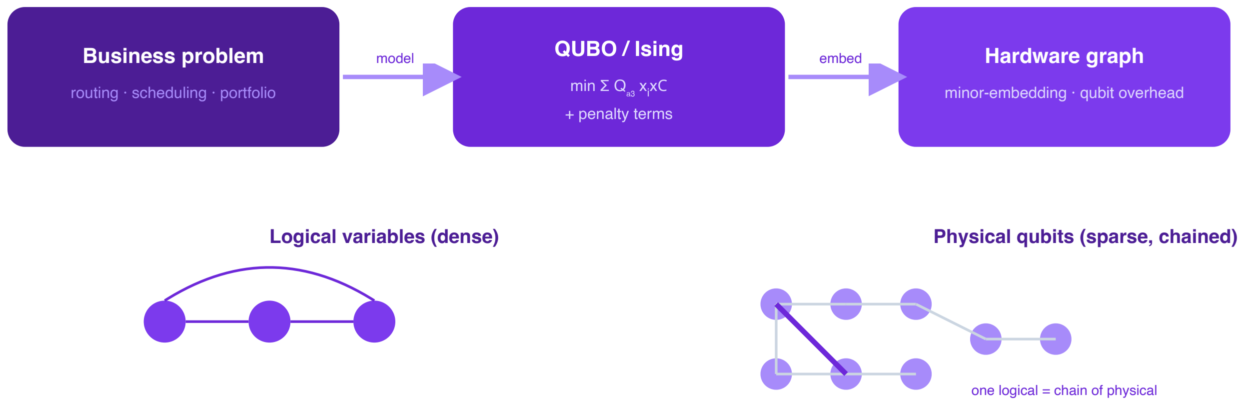

Figure 4.1 — The mapping pipeline. A business problem becomes a QUBO/Ising model with penalty terms, then must be embedded onto a sparse hardware graph — where each densely-connected logical variable may require a chain of physical qubits.

The first obstacle is that real problems have **constraints** — a vehicle must visit each stop exactly once, a portfolio must be fully invested, a schedule must respect rest rules — but QUBO is *unconstrained* by definition. Constraints are therefore folded into the objective as **penalty terms**: a large penalty weight is added so that violating a constraint costs more energy than any feasible solution. Choosing these weights is delicate. Too small, and the solver returns infeasible answers; too large, and the feasibility penalties swamp the actual objective, flattening the landscape and making the optimum hard to find. This penalty tuning is a genuine engineering burden with no universal recipe.

```
# Encoding "visit each city exactly once" as a penalty
# x[i][t] = 1 if city i is visited at step t
H_cost    = sum(dist[i][j] * x[i][t] * x[j][t+1] for i,j,t)
H_oneCity = sum((1 - sum(x[i][t] for t))**2 for i)   # each city once
H_oneStep = sum((1 - sum(x[i][t] for i))**2 for t)   # one city per step
Q = H_cost + LAMBDA * (H_oneCity + H_oneStep)         # LAMBDA = penalty weight
```

Listing 4.1 — Constraints become squared penalty terms scaled by a weight LAMBDA. Note the quadratic blow-up: an n-city tour needs n^2 binary variables before any hardware mapping.

The second obstacle, specific to hardware, is **minor-embedding**. A QUBO defines a graph of variable interactions, but a real quantum processor has fixed, sparse physical connectivity — each qubit couples to only a handful of neighbours. To run a densely-connected problem, several physical qubits must be chained together to act as one logical variable. This embedding multiplies the qubit requirement substantially; a problem with a few hundred densely-coupled logical variables can exhaust a several-thousand-qubit annealer. The overhead is one of the principal reasons the impressive raw qubit counts in marketing materials do not translate into correspondingly large solvable problems.

Together, penalty encoding and embedding explain a sobering truth: the path from a real logistics or portfolio problem to something a quantum device can attempt is long, lossy, and expensive in qubits. Quantum-inspired solvers share the QUBO formulation but skip the embedding step entirely, which is one more reason they are often the more practical destination for the same modelling effort.

SECTION FIVE

5. Where Advantage Is — and Isn't

This is the section a candid whitepaper owes its reader. Stripped of marketing, what can quantum optimization actually do for a logistics or finance operation in 2026, and what can it not?

Begin with what it cannot do. There is, at the time of writing, **no robust, peer-reviewed demonstration of a scaling quantum advantage** for industrial-scale routing, scheduling, or portfolio optimization over the best classical methods. The barriers are concrete. Gate-model machines have too few high-quality qubits — tens to low hundreds of usable qubits, far short of the thousands an embedded industrial QUBO would demand. Gate **noise** limits circuit depth, so QAOA cannot run enough layers to reach the regime where theory predicts benefit. **Connectivity** is sparse, inflating embedding cost. And annealers, while large, have not shown a consistent edge over well-engineered classical solvers in independent tests. These are not pessimistic opinions; they are the consensus of the research literature.

> **REALITY CHECK**
>
> "Quantum supremacy" has been demonstrated only for contrived sampling tasks chosen because they are hard for classical computers and have no practical use. It has **not** been demonstrated for any commercially meaningful optimization problem. Treat any 2026 claim of present-day quantum advantage on routing, scheduling, or portfolios as marketing until you have seen the classical baseline and an independent reproduction.

Now the honest positive case. The first place value is real today is **quantum-inspired classical optimization**. Digital annealers, GPU-accelerated Ising solvers, and tensor-network methods take the QUBO modelling discipline that quantum research has popularised and run it at large scale on classical silicon. Organisations have used these to attack QUBO-formulated routing and portfolio problems and obtained competitive results against traditional solvers — and, importantly, the modelling skills transfer directly to quantum hardware if and when it matures. The value here is not "quantum"; it is the combination of a clean problem formulation and fast classical hardware.

The second place value is emerging is **hybrid methods on small or structured sub-problems**. Some workflows decompose a large problem and hand a small, hard core to a quantum or hybrid solver while classical methods handle the rest. For small instances this is a legitimate research avenue and a way to build organisational capability, but it is not yet a source of production advantage at enterprise scale.

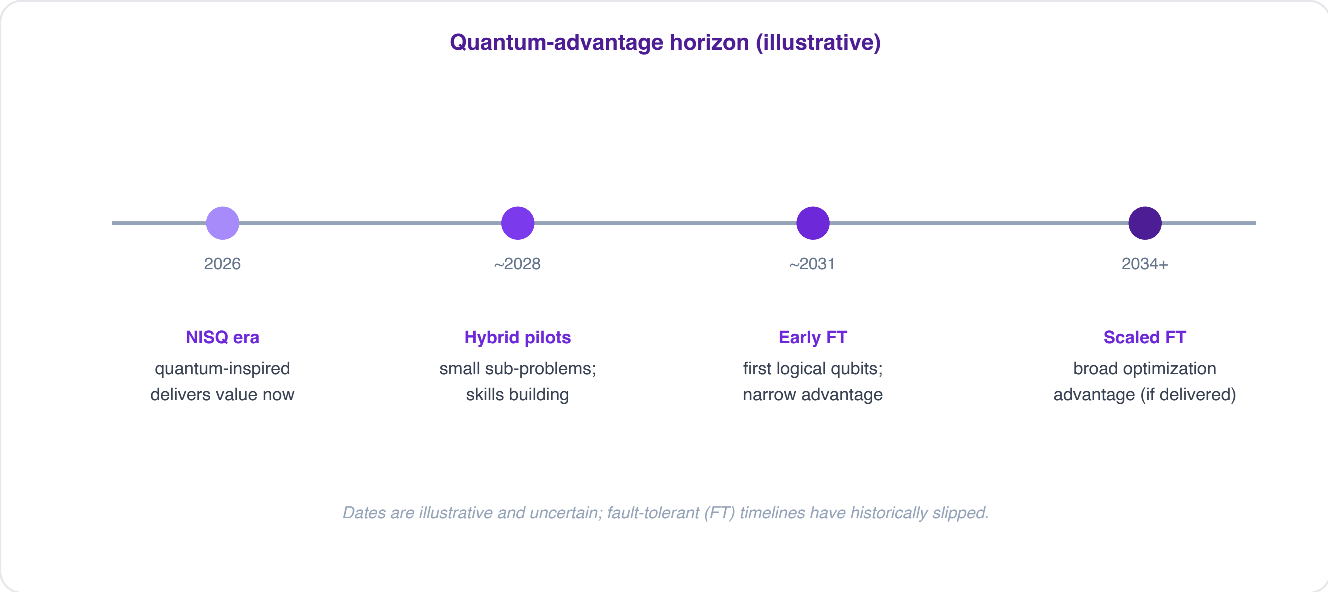

Figure 5.1 — A realistic horizon. Near-term value is quantum-inspired and hybrid; broad fault-tolerant advantage for these problems is years away and far from guaranteed.

The realistic horizon is therefore staged. Through the late 2020s, the deployable technology is quantum-inspired and classical. The arrival of early fault-tolerant machines — a few logical qubits — may unlock narrow advantages on specially structured problems early in the next decade. Broad, dependable advantage across routing, scheduling, and portfolio optimization at industrial scale, if it arrives at all, is a 2030s prospect contingent on error-correction breakthroughs that have not yet happened. Prudent enterprises plan for this horizon by building QUBO modelling capability now and treating quantum hardware as an option to be ready for, not a system to bet the operation on.

SECTION SIX

6. Methodology: Benchmark Setup

To compare approaches fairly one needs well-defined problem families, credible classical baselines, and transparent metrics. This section states the setup so the reader understands exactly what the Section 7 charts do and do not claim.

We study three problem families representative of the introduction. **Capacitated vehicle routing (CVRP)** instances range from 20 to 200 customers with vehicle capacity and depot constraints. **Job-shop scheduling** instances range from 10 to 100 jobs across a fixed set of machines. **Cardinality-constrained portfolio optimization** instances range from 30 to 300 assets with an integer asset-count limit. Each family is parameterised so that problem size — the variable on the horizontal axis of our charts — has a clear, monotone meaning.

The **classical baselines** are deliberately strong, because a benchmark against a weak baseline is worthless. We use state-of-the-art mathematical-programming and metaheuristic solvers of the kind an experienced operations-research team would actually deploy: branch-and-cut mixed-integer solvers for exact comparison where tractable, and high-quality metaheuristics (large-neighbourhood search, tabu search) for larger sizes. The **quantum-inspired** arm is a GPU-accelerated simulated/digital-annealing solver running the same QUBO formulation. The **QAOA** arm represents gate-model performance using shallow circuits, reflecting NISQ-era depth limits.

Benchmark configuration (illustrative)

Dimension	Setting	Dimension	Setting
Problem families	CVRP · job-shop · portfolio	Size range	20–300 variables
Classical baseline	MIP + metaheuristics	Quantum-inspired	GPU Ising / digital anneal
Quantum arm	QAOA, shallow depth	Time budget	Fixed wall-clock / instance
Primary metric	Solution-quality gap (%)	Secondary	Time-to-solution

We report two metrics. **Solution quality** is expressed as the percentage gap from the best known (or proven optimal, where available) objective value — lower is better. **Time-to-solution** is the wall-clock time to reach a target quality, capturing the practical question of whether a method is fast enough to be useful. Reporting both matters because a method that finds excellent solutions too slowly to act on, or fast solutions of poor quality, is not useful regardless of which paradigm produced it.

IMPORTANT CAVEAT

The charts in Section 7 are **modeled and illustrative**. They are constructed to convey the *structure* of the comparison — the shape of the curves and the rank-order of the methods at different sizes — consistent with the published literature and our own engineering experience. They are **not** measurements from a specific machine on a specific day, and they should never be cited as a vendor benchmark.

We also fix the time budget per instance so the comparison is honest: every method is given the same wall-clock allowance, because an unbounded-time comparison flatters whichever method is allowed to run longest. A subtle but important point concerns **where the clock starts and stops**. For NISQ methods, the wall-clock cost is not only the time the quantum processor spends executing a circuit; it includes the classical optimisation loop that proposes new circuit parameters, the compilation and embedding of the problem onto the hardware graph, and — on shared cloud machines — the queue time before a job runs at all. A benchmark that reports only the quantum-processing-unit time, ignoring this surrounding overhead, paints a misleadingly favourable picture. We therefore measure end-to-end wall-clock, which is the only figure an operations team can actually act on.

Finally, a word on what we deliberately exclude. We do not model financial cost-per-run, which varies enormously between owned classical hardware and metered quantum-cloud access and would conflate a technology comparison with a pricing one. We do not model the engineering effort to formulate each problem as a QUBO, which is substantial but one-time and shared across all QUBO-consuming methods. And we do not extrapolate to fault-tolerant hardware, because doing so would require assuming error-correction milestones that have not been demonstrated. The comparison is intentionally narrow and present-tense, which is what makes it honest. With the setup stated, the findings follow.

SECTION SEVEN

7. Findings

Three patterns recur across the modeled families: quantum-inspired methods are competitive with strong classical solvers across all sizes, NISQ-era QAOA degrades as problems grow, and time-to-solution favours mature classical and quantum-inspired tooling today.

~300	<30	0	2030s
VARIABLES TRACTABLE, QUANTUM-INSPIRED	VARIABLES BEFORE QAOA DEGRADES	ROBUST INDUSTRIAL QUANTUM-ADVANTAGE RESULTS	REALISTIC FAULT-TOLERANT HORIZON

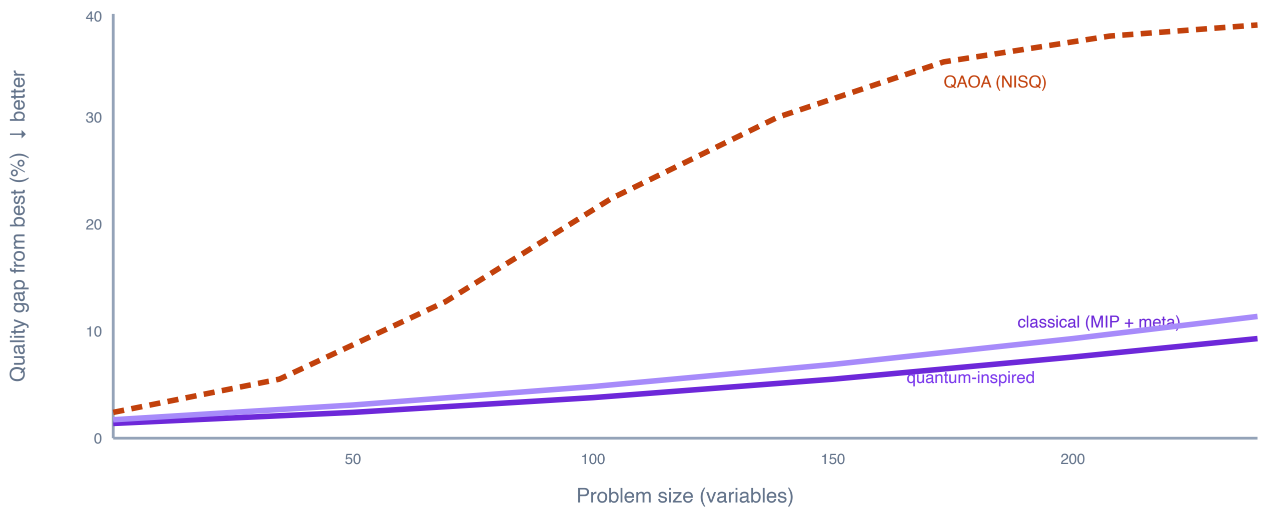

Figure 7.1 — Solution quality vs problem size (lower is better). Quantum-inspired tracks strong classical solvers across the range; NISQ-era QAOA is competitive only at very small sizes and degrades sharply as circuits deepen.

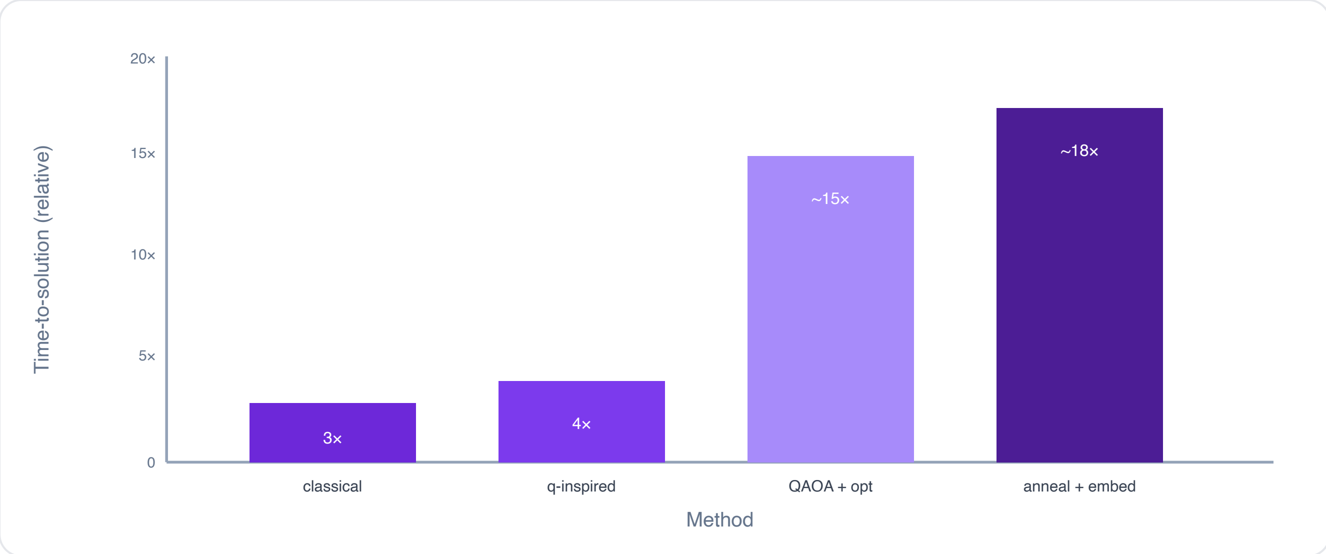

Figure 7.2 — Time-to-solution to reach a target quality (relative, illustrative). The classical-optimiser and embedding/queue overheads of NISQ methods dominate their wall-clock cost today.

The quality chart (Figure 7.1) is the central result. Across the modeled range, the quantum-inspired solver shadows the strong classical baseline closely — neither dramatically better nor worse — which is exactly why quantum-inspired methods are the deployable choice: they match best-in-class quality on hardware that is available today. The QAOA curve tells the cautionary tale: competitive on toy sizes, it deteriorates rapidly as problem size forces deeper circuits that NISQ noise cannot sustain. This is the gap between theoretical promise and present hardware made visible.

The time chart (Figure 7.2) reflects a practical reality that quality charts often hide: the wall-clock cost of NISQ methods is dominated by overhead the quantum step does not pay for. QAOA's classical optimisation loop requires many noisy circuit evaluations to estimate each cost gradient; annealing requires minor-embedding the problem and waiting in a shared-machine queue. Mature classical and quantum-inspired tooling simply returns answers faster today, because decades of solver engineering and the absence of an embedding step give them a structural head start that the quantum step does not currently overcome.

Read together, the two charts deliver a consistent message. On the axis enterprises care about most — good answers, fast enough to act on — quantum-inspired methods are competitive with the best classical solvers across the whole modeled range, while NISQ-era quantum hardware is competitive only on small instances and pays a heavy time penalty even there. Nothing in these results forecloses a future quantum advantage; they simply locate it firmly in the future. The pattern also explains why so many organisations that began a "quantum" optimization project find their durable wins came from the QUBO formulation and the classical or quantum-inspired solver they built around it, rather than from the quantum hardware that motivated the effort. That is not a failure of the project; it is the field as it honestly stands in 2026.

RESULT SUMMARY

- Quantum-inspired methods match strong classical solvers on solution quality across the modeled sizes — the deployable advantage is the formulation, not the hardware.
- NISQ-era QAOA is competitive only below roughly 30 variables and degrades sharply as required circuit depth grows.
- Time-to-solution favours classical and quantum-inspired tooling; NISQ overhead (classical loops, embedding, queues) dominates wall-clock cost.
- No robust, reproduced industrial quantum-advantage result exists for these problem families as of 2026.

SECTION EIGHT

8. Discussion, Limitations & Roadmap

The findings point to a clear, unglamorous conclusion: the optimization value an enterprise can capture today comes from disciplined problem formulation and fast classical hardware, with quantum hardware as a capability to prepare for rather than depend on.

Limitations. These results are modeled and illustrative, constructed to reflect the structure reported in the published literature and our engineering experience rather than measurements from a particular machine. Real quantum hardware is improving — qubit counts, coherence times, and error rates all advance year over year — so a snapshot in 2026 is exactly that. The QAOA arm represents shallow NISQ circuits; theoretical results suggest deeper, error-corrected QAOA could behave very differently, but that regime is not available today. We have also deliberately bounded scope to optimization; quantum computing's strongest theoretical case lies elsewhere (notably quantum chemistry and certain cryptographic problems), and those should not be conflated with optimization advantage.

Risks of over-investment. The dominant risk we observe is misallocated investment driven by hype: committing capital to production quantum hardware on the expectation of a near-term advantage that the evidence does not support. The opposite risk — ignoring the field entirely and lacking the skills to adopt it when it matures — is real but smaller and cheaper to mitigate. The balanced posture is to invest in people and formulations, not in betting the operation on present-day quantum throughput.

> IN PRACTICE
>
> The single most valuable asset an enterprise can build now is the ability to formulate its hard problems cleanly as QUBO/Ising models. That skill pays off immediately on quantum-inspired and classical solvers, and it is exactly the skill needed to exploit quantum hardware the day it becomes advantageous. The formulation is the durable investment; the hardware underneath it is replaceable.

Recommendations

1. **Pilot quantum-inspired solvers now.** GPU Ising and digital-annealing solvers deliver real value today on routing, scheduling, and portfolio problems already cast as QUBOs — and the modelling effort transfers to quantum hardware later.
2. **Invest in QUBO/Ising formulation skills,** not in production quantum hardware. The formulation is the durable asset; treat the hardware layer as interchangeable.
3. **Run small gate-model and annealing experiments to learn,** not to deploy. Use them to build organisational fluency and to be ready, not as systems the operation relies upon.

4. **Demand honest baselines.** Accept no quantum-advantage claim without a strong, well-tuned classical comparison and, ideally, independent reproduction.
5. **Plan on a multi-year horizon.** Budget for fault-tolerant advantage in the 2030s, treat earlier dates as optimistic, and revisit the assessment annually as hardware advances.

Conclusion

Quantum computing is a genuine and important long-term technology, and optimization is a natural target for it. But in 2026 the honest assessment is that quantum hardware does not yet beat strong classical methods on the routing, scheduling, and portfolio problems that matter to logistics and finance. The deployable value today is quantum-*inspired* — clean problem formulation run on fast classical silicon — with hybrid experiments to build skills and a fault-tolerant horizon measured in years. Enterprises that invest in formulation capability and resist the hype will be precisely the ones positioned to capture real quantum advantage the day it arrives. Quantum value should be measured, not hyped.

References

1. Preskill, J. — Quantum Computing in the NISQ era and beyond, *Quantum* 2 (2018).
2. Farhi, E., Goldstone, J., Gutmann, S. — A Quantum Approximate Optimization Algorithm (2014).
3. Lucas, A. — Ising formulations of many NP problems, *Frontiers in Physics* (2014).
4. Cerezo, M. et al. — Variational Quantum Algorithms, *Nature Reviews Physics* (2021).
5. King, A. D. et al. — Quantum annealing benchmarking and quantum-inspired comparisons (D-Wave, various).
6. Root Digit Quantum Lab — Internal QUBO benchmarking methodology and modeling notes (2026).

Quantum value — measured, not hyped.

Root Digit's Quantum Lab helps logistics and finance teams capture optimization value that is real today — clean QUBO/Ising formulations, quantum-inspired solvers on classical hardware, and a clear-eyed roadmap for the quantum hardware that is coming. We separate the signal from the marketing, so your investment lands where the evidence supports it.

Talk to our quantum team →

rootdigit.com

General: info@rootdigit.com · **Consultation:** rootdigit.com/contact/consultation
Dubai International Financial Centre (DIFC) · Wyoming, USA