

Introduction

Optimization is the art of making the best possible choice under limits.

That sentence sounds simple, but it describes a huge part of modern life. A delivery company wants to choose routes that reduce fuel cost. A hospital wants to schedule nurses while respecting rest periods and patient needs. An investment manager wants to choose a portfolio that balances expected return and risk. A factory wants machines, workers, and materials to move in the right order. A machine-learning engineer wants model parameters that reduce prediction error. In each case, there are many possible choices, and we want one that is “best” according to some rule.

That rule is called an objective function. An objective function assigns a number to each possible solution. If we are minimizing delivery distance, the objective function might output the total number of kilometers traveled. If we are maximizing profit, it might output expected profit. The limits are called constraints. A delivery truck cannot carry more than its capacity. A worker cannot be scheduled for two jobs at the same time. A portfolio may be required to include no more than a certain number of assets.

So, at the most basic level:

> Optimization means searching through possible solutions to find one with the best objective value while satisfying the required constraints.

This book is about a particular family of methods for optimization: variational quantum algorithms, often abbreviated as VQAs. These are hybrid algorithms that use both a quantum computer and a classical computer. The quantum computer prepares and measures quantum states. The classical computer adjusts parameters, checks results, and decides what to try next. VQAs became especially important because they are designed for the current era of quantum hardware, often called the noisy intermediate-scale quantum era, or NISQ era, a term popularized by John Preskill to describe devices with many physical limitations: not yet error-corrected, affected by noise, but still large enough to explore meaningful experiments (Preskill, 2018).

This introduction gives you the first orientation. We will not assume that you already know quantum mechanics, advanced optimization, or computer science theory. We will build those ideas carefully in later chapters. For now, the goal is to understand what kind of journey this book will take.

Why optimization becomes difficult

Some optimization problems are easy to state but hard to solve.

Imagine you have 20 tasks and 20 time slots. You want to assign each task to one slot while minimizing delay and avoiding conflicts. Even before adding realistic constraints, the number of possible assignments can grow extremely fast. This kind of rapid growth is called combinatorial explosion.

“Combinatorial” means related to combinations or arrangements.

“Explosion” means that the number of possibilities increases so quickly that checking them one by one becomes unrealistic.

For example, if you must choose whether to include or exclude each of 50 assets in a portfolio, there are

2^{50}

possible yes-or-no selections. That is more than one quadrillion possibilities. A modern computer cannot simply inspect every option in many such problems, especially when the objective function and constraints are complicated.

This does not mean classical computers are weak. Classical optimization is a mature and powerful field. Methods such as linear programming, integer programming, branch and bound, simulated annealing, local search, and modern heuristics solve many real industrial problems very well. A serious quantum optimization study must compare against these strong classical methods, not against weak examples. Later in the book, Chapter 18 will focus on this point: quantum optimization must be benchmarked honestly.

But the difficulty of many optimization problems leaves room for research. Some problems have structures that may be hard for classical methods to exploit fully. Quantum algorithms offer different ways to represent and explore possible solutions. Whether those differences produce practical advantage depends on the problem, the hardware, the encoding, the algorithm, the noise level, and the classical baseline used for comparison.

The honest message is:

> Quantum optimization is promising, but it is not magic. It is a research area with potential, limitations, and many open questions.

What makes a method “quantum”?

A quantum computer processes information using quantum systems. The basic unit of quantum information is a qubit, short for “quantum bit.” A classical bit has value 0 or 1. A qubit, before measurement, can be described by a quantum state involving amplitudes for 0 and 1. These amplitudes are not ordinary probabilities, although they are related to probabilities when we measure.

Quantum algorithms use features such as superposition, interference, and entanglement.

A superposition is a quantum state that is not simply one classical alternative. For example, a qubit can be in a state involving both 0 and 1 amplitudes before measurement. This does not mean we can directly read both answers at once. Measurement gives one outcome according to probability rules. The power of quantum algorithms comes from how amplitudes are transformed before measurement.

Interference means that amplitudes can combine in ways that strengthen some outcomes and weaken others. A useful quantum algorithm tries to increase the probability of good answers and decrease the probability of bad answers.

Entanglement is a kind of correlation between quantum systems that cannot be explained as each qubit simply carrying its own independent state. Entanglement is one of the main reasons quantum circuits can represent certain states differently from classical descriptions.

These ideas will be developed carefully in Chapter 4 and Chapter 5. For now, it is enough to know that quantum optimization algorithms try to use quantum state preparation and measurement to produce information about good solutions.

What does “variational” mean?

The word variational comes from a long tradition in mathematics and physics. In a variational method, we choose a family of possible candidates and then vary adjustable parameters to improve them.

A simple classical example is fitting a line to data. Suppose we choose lines of the form

$$y = ax + b.$$

The parameters are a and b . We vary a and b until the line fits the data well. The family of candidates is “all lines of this form,” and the objective function might be the average squared error between predictions and true values.

A variational quantum algorithm uses the same broad idea, but the candidate is a quantum state prepared by a parameterized quantum circuit. A parameterized quantum circuit is a quantum circuit containing adjustable numbers, such as rotation angles. The quantum computer prepares a state using those parameters. Then we measure the state to estimate a cost. A classical optimizer uses that cost to choose better parameters. This loop repeats.

The basic structure is:

1. Choose initial parameters.
2. Run a quantum circuit using those parameters.
3. Measure the output many times.
4. Estimate the cost.
5. Use a classical optimizer to update the parameters.
6. Repeat until the result stops improving or the budget is used.

This hybrid structure is central to the book. It is one reason VQAs are considered suitable for NISQ devices: they can use relatively short quantum circuits while leaving much of the decision-making to classical computation. Reviews of VQAs describe them as a major framework for near-term quantum algorithms, while also emphasizing challenges such as noise, trainability, and benchmarking (Cerezo et al., 2021).

Two famous examples: VQE and QAOA

Two algorithms will appear often in this book: the Variational Quantum Eigensolver and the Quantum Approximate Optimization Algorithm.

The Variational Quantum Eigensolver, or VQE, was originally developed for estimating low-energy states of quantum systems, especially in quantum chemistry. In chemistry, the energy of a molecule is connected to its electronic structure. VQE uses a parameterized quantum circuit to prepare trial quantum states, then estimates their energy. The classical optimizer adjusts the parameters to reduce the energy. The early experimental demonstration by Peruzzo and collaborators helped establish VQE as a practical model for hybrid quantum-classical algorithms (Peruzzo et al., 2014).

The Quantum Approximate Optimization Algorithm, or QAOA, was proposed for combinatorial optimization problems such as MaxCut, where the goal is to cut a graph into two parts while maximizing the number of edges crossing between the parts (Farhi et al., 2014). QAOA alternates between operations related to the problem's objective and operations that mix possible solutions. It has a depth parameter, usually called p , which controls how many alternating layers are used. Larger p can make the circuit more expressive, but it also usually requires more quantum gates, which can be harder for noisy hardware.

VQE and QAOA are not the only VQAs, but they are excellent teaching examples. VQE shows how variational methods connect to physics and chemistry. QAOA shows how they connect to discrete optimization and real-world decision problems.

From real problems to binary variables

Many quantum optimization methods work most naturally with binary variables. A binary variable is a variable that can take one of two values, often 0 or 1.

For example, suppose we are choosing which projects to fund:

$$x_i = \begin{cases} 1, & \text{if project } i \text{ is selected,} \\ 0, & \text{if project } i \text{ is not selected.} \end{cases}$$

A portfolio selection problem can be written similarly:

$$x_i = \begin{cases} 1, & \text{if asset } i \text{ is included,} \\ 0, & \text{if asset } i \text{ is excluded.} \end{cases}$$

Once a problem is written in binary variables, it is often possible to express it as a QUBO problem. QUBO means quadratic unconstrained binary optimization. "Quadratic" means the objective includes terms involving one variable, such as x_i , and pairs of variables, such as $x_i x_j$. "Unconstrained" means constraints have been absorbed into the objective, often using penalty terms. "Binary" means the variables are 0 or 1.

A typical QUBO has the form

$$\min_x \sum_i a_i x_i + \sum_{i < j} b_{ij} x_i x_j,$$

where each x_i is either 0 or 1. The numbers a_i and b_{ij} define the cost of selecting individual items and pairs of items.

A closely related form is the Ising model, which uses variables $s_i \in \{-1, +1\}$ instead of $x_i \in \{0, 1\}$. The Ising model originally comes from statistical physics, but it is widely used as a mathematical form for optimization. Many NP-hard problems can be expressed as Ising formulations, as surveyed by Lucas (2014). Later, Chapter 11 will explain carefully how to move between real problem descriptions, QUBO models, and Ising Hamiltonians.

This translation step is crucial. A quantum algorithm does not automatically understand “make my supply chain better.” We must first build a mathematical model. If the model is poor, the quantum algorithm may solve the wrong problem very efficiently.

Where quantum optimization may be applicable

The learner’s main question for this book is practical: Where can quantum optimization techniques apply in real life?

The possible application areas include:

- logistics routing,
- supply-chain planning,
- energy-grid optimization,
- financial portfolio construction,
- drug discovery,
- materials science,
- telecommunications,
- manufacturing,
- scheduling,
- machine-learning model selection.

But “applicable” has several meanings.

A method can be conceptually applicable if the problem can be written in a form that a quantum optimization algorithm can process. For example, a simplified delivery routing problem may be expressible as a QUBO.

A method can be experimentally applicable if we can run small versions on a simulator or current quantum hardware and learn something from the result.

A method can be commercially useful only if it performs well enough on real problem sizes, with real constraints, compared with strong classical alternatives, at acceptable cost and reliability.

These are different levels. Much of today's quantum optimization work is in the first two categories: model building, small-scale experiments, hybrid workflows, and benchmarking. The third category—broad, reliable practical advantage—is still an open goal for most optimization applications on current quantum hardware.

This distinction matters throughout the book. We will discuss real-world domains, but we will not pretend that every problem is already solved by quantum computers. A good quantum optimization practitioner must know both the opportunity and the boundary.

What this book will teach you

This book is organized as a gradual path.

First, we study optimization itself. You will learn objective functions, constraints, vectors, matrices, gradients, local optima, global optima, convexity, discrete search spaces, and classical optimization methods. This matters because quantum optimization is still optimization. Without the classical foundation, the quantum part becomes a collection of mysterious circuit diagrams.

Then we build quantum computing from first principles. You will learn what qubits are, how quantum states are represented, how measurement works, and how gates become circuits. You will also learn why quantum measurement gives samples rather than a direct full answer.

After that, we enter the variational framework. You will learn how parameterized circuits are designed, how cost functions are measured, how classical optimizers update parameters, and why noise makes this process difficult.

Then we study major algorithms and encodings: VQE, QUBO, Ising models, QAOA, constrained variants, warm-start methods, recursive approaches, and quantum annealing. We will connect these methods to canonical problems such as MaxCut, graph coloring, independent set, portfolio selection, knapsack, and scheduling.

Finally, we move toward practice. We will discuss software tools, real-life application areas, case studies, benchmarking, and project design. The goal is not only to know definitions, but to be able to ask responsible questions such as:

- Can this real problem be written as a QUBO or Hamiltonian?
- What constraints are essential, and how should they be encoded?
- How many variables and interactions does the model require?
- What classical baseline should be used?
- How much sampling is needed to estimate the cost reliably?
- Does the quantum method improve solution quality, runtime, robustness, or insight?
- Is the experiment a proof of concept, or evidence of practical advantage?

These questions are more valuable than hype. They are the habits of careful scientific and engineering work.

A first small example

Suppose a small company can choose among three projects: A, B, and C. Each project has a benefit and a cost.

Project	Benefit	Cost
A	8	5
B	6	4
C	5	3

The company has a budget of 7. It wants the highest total benefit without exceeding the budget.

We can define binary variables:

$$x_A, x_B, x_C \in \{0, 1\}.$$

If $x(A) = 1$, project A is selected. If $x(A) = 0$, it is not selected.

The benefit is

$$8x_A + 6x_B + 5x_C.$$

The cost constraint is

$$5x_A + 4x_B + 3x_C \leq 7.$$

Checking the possibilities:

- Choose A only: cost 5, benefit 8.
- Choose B only: cost 4, benefit 6.
- Choose C only: cost 3, benefit 5.
- Choose A and B: cost 9, not allowed.
- Choose A and C: cost 8, not allowed.
- Choose B and C: cost 7, benefit 11.
- Choose all three: cost 12, not allowed.

The best valid choice is B and C, with benefit 11.

This tiny problem does not need a quantum computer. It is easy to solve by hand. But it shows the modeling pattern that appears in larger real problems:

1. Define decision variables.
2. Define the objective.
3. Define constraints.
4. Search for the best valid solution.
5. If using quantum optimization, encode the problem into a form the algorithm can process.

Real industrial versions may involve thousands or millions of variables, uncertain data, multiple objectives, time-dependent constraints, and changing conditions. The simple pattern remains, but the scale and modeling difficulty grow.

The attitude of this book

This book takes a balanced position.

We will study quantum optimization because it is intellectually important and potentially useful. Quantum computing gives us a new computational model, and variational algorithms offer a practical way to test ideas on imperfect hardware.

At the same time, we will avoid exaggerated claims. Current quantum devices are noisy. Measurements are statistical. Variational landscapes can be hard to train. Encoding real problems can introduce many extra variables. Classical optimization is very strong. A small quantum demonstration is not the same as practical advantage.

The right attitude is curiosity with discipline.

Curiosity asks: Can quantum mechanics help us search, sample, or represent solutions in useful ways?

Discipline asks: Compared with what? At what scale? With what error? Under what assumptions?

If you keep both questions alive, you will read this book well.

References

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3, 625–644. <https://doi.org/10.1038/s42254-021-00348-9>

Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv:1411.4028*. <https://arxiv.org/abs/1411.4028>

Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in Physics*, 2, Article 5. <https://doi.org/10.3389/fphy.2014.00005>

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., & O'Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5, Article 4213. <https://doi.org/10.1038/ncomms5213>

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, Article 79. <https://doi.org/10.22331/q-2018-08-06-79>

Document information

Introduction

Project	Variational Quantum Algorithms for Optimization
Document	Document 1.4
Author	phone
Verifier	Not verified
Downloaded	July 03, 2026 17:56 KST
Status	Working
Document link	https://theorytrace.com/projects/variational-quantum-algorithms-for-optimization/documents/introduction/