qubit-lab.ch

QAOA RQP
Rapid Quantum Prototype for Portfolio Optimization

Quick guide for using the Excel-to-Quantum optimization tool

Prepared for external testers and discussion partners
Version 1.0 | April 2026

One data definition can be used to generate the optimization model, circuit setup, and comparable output tables.

Purpose of this guide

This document explains how to use the QAOA RQP tool as a rapid quantum prototyping environment. It is written for business, finance, and technology users who want to explore quantum optimization without first becoming quantum-computing specialists.

1. Introduction

qubit-lab.ch's mission is to make quantum computing accessible to organizations and professionals who are not quantum physicists. The focus is practical: define a business problem, translate it into a quantum-ready formulation, run controlled experiments, and understand what the results mean.

QAOA RQP is a Rapid Quantum Prototype for portfolio optimization. It lets users describe a portfolio selection problem in an Excel workbook and then use an online or offline tool to build the optimization model and run the corresponding simulation setup.

What QAOA RQP is

A prototyping tool for learning, testing, and structured discussion. It is not an investment recommendation engine, not a production portfolio management system, and not a claim that quantum computers currently outperform classical optimizers for this task.

2. Quick start

1. Open the provided Excel workbook and review the Assets and Settings sheets.
2. Define the selectable portfolio options. Each variable option becomes one binary decision variable and therefore one qubit in the QUBO/QAOA formulation.
3. Optionally mark existing positions as fixed. Fixed positions are included in every portfolio, but they do not become quantum decision variables.
4. Set model parameters such as budget, risk-free rate, budget penalty, variance penalty, QAOA layers, iterations, and restarts.
5. Upload the workbook in the online tool or run the offline version.
6. Choose classical_only for a classical benchmark or qaoa_limited for the controlled quantum simulation path.
7. Run the optimization and review the result summary, candidates, portfolio contents, charts, circuit overview, and diagnostic logs.

Workbook example: selectable asset options

decision_id	Ticker	Decision Type	Company	Asset Group	Option Label	Shares	Approx Cost USD
NVDA_opt1	NVDA	fixed	NVIDIA	NVDA	100k	498	100,041
AAPL_opt1	AAPL	fixed	Apple	AAPL	100k	372	100,034
MSFT_opt1	MSFT	variable	Microsoft	MSFT	100k	236	100,144
AVGO_opt1	AVGO	variable	Broadcom	AVGO	100k	248	99,994
MU_opt1	MU	variable	Micron Technology	MU	100k	218	99,997
ORCL_opt1	ORCL	variable	Oracle	ORCL	100k	550	99,910
AMD_opt1	AMD	variable	Advanced Micro Device	AMD	100k	361	99,903

3. What the tool builds

From the workbook, QAOA RQP builds a complete experimental chain:

- a binary portfolio decision model, where each variable asset option is represented by one bit;
- a QUBO formulation that combines return, risk, and budget terms;
- an Ising conversion suitable for quantum-circuit construction;
- a QAOA circuit setup with configurable layers, iterations, restarts, and sampling settings;
- a simulation run, subject to memory, runtime, and license limits;
- structured outputs for classical candidates, quantum samples, solver comparison, portfolio contents, and diagnostics.

Excel workbook → **QUBO model** → **Ising conversion** → **QAOA circuit** → **Simulation** → **Results export**

One data definition can be used to generate the optimization model, circuit setup, and comparable output tables.

4. Workbook structure

The Excel workbook is the main user interface for defining the problem. The example workbook contains both input sheets and output sheets. The most important input sheets are:

Sheet	Purpose	Typical user action
Assets	Defines selectable options and fixed positions.	Add or adjust rows for tickers, option labels, shares, approximate costs, and decision type.
Settings	Defines model and solver parameters.	Set budget, risk-free rate, penalty weights, QAOA layers, iterations, restarts, and export settings.
Returns	Provides return and volatility inputs by ticker.	Review or update expected return and volatility assumptions.
Covariance / AnnualizedCovariance	Provides risk interaction terms across tickers.	Review or replace covariance assumptions if using your own data.
Results_* / QAOA_*	Stores generated outputs after a run.	Review outputs; do not normally edit manually.

Workbook example: model and solver settings

Key	Value	Description
budget_usd	1'000'000	Target total budget
risk_free_rate_annual	0.04	Annual risk-free rate used in excess-return reward
lambda_budget	50	Budget deviation penalty
lambda_variance	6	Variance contribution weight
	0	Unused in current notebook; kept for transparency
top_n_export	20	Number of candidate portfolios exported to overview :
	0	
enable_qaoa	1	1 = run QAOA, 0 = skip QAOA
qaoa_p	6	QAOA depth / number of layers
qaoa_maxiter	100	Optimizer iteration budget for QAOA

5. Fixed and variable positions

Variable positions are options that the optimizer may select or reject. Each variable option becomes one binary decision variable and therefore one qubit. **Fixed positions** represent existing holdings or mandatory positions. They are included in every portfolio and affect the total budget, risk, and return, but they do not increase the qubit count.

Practical implication

A portfolio can include many existing fixed holdings while keeping the quantum problem smaller. The qubit count is driven by the number of variable options, not by every line in the portfolio workbook.

6. Running the tool

The online tool follows a simple workflow:

1. Enter the license key.
2. Upload the Excel workbook.
3. Inspect the workbook summary and pre-run runtime estimate.
4. Select the mode: classical_only or qaoa_limited.
5. Set the response level. Compact is faster and lighter; full returns more diagnostics and result tables.
6. Choose QAOA settings where relevant: layers, iterations, restarts, warm start, and shots.
7. Run the optimization and wait for the result summary.

One active run per key. Each license key is designed for one active simulation run at a time. If a second run is started with the same key before the first run has finished, the tool rejects the second request with a clear message. This prevents accidental duplicate compute usage and keeps capacity predictable.

7. Interpreting outputs

The result view is designed to compare classical and quantum-derived candidates in a consistent format. Depending on response level and mode, the tool may return:

- Classical Result Summary: best classical candidate, QUBO value, selected amount, budget gap, return proxy, volatility, and Sharpe-like score.
- Quantum Result Summary: best quantum/QAOA-derived candidate, probability, QUBO value, selected amount, budget gap, return proxy, volatility, and Sharpe-like score.
- Top Classical Candidates: ranked candidate bitstrings from the classical candidate pool.
- Top Quantum Candidates: ranked sampled or exported QAOA candidates.
- Portfolio Contents: selected assets and option blocks for a candidate portfolio.
- Offline-style Charts: risk/return views, QUBO breakdowns, optimization history, and circuit overview where available.
- Circuit Overview: estimated qubit count, layers, sequential two-qubit depth, total gates, and related circuit metrics.

Example output: ranked portfolio candidates

rank	bitstring	source	probability	qubo_value	return_term	risk_term
1	11111000010000	classical_heuristic	nan	0.5702	-0.2628	0.8330
2	11110000011000	classical_heuristic	nan	0.5829	-0.1509	0.7337
3	11110000010010	classical_heuristic	nan	0.5865	-0.1474	0.7340
4	11101000010010	qaoa_full_pennylane_p6	0.000292	0.5905	-0.2746	0.8652
5	11110001010000	classical_heuristic	nan	0.5911	-0.1886	0.7797
6	11101000011000	classical_heuristic	nan	0.5965	-0.2780	0.8745

8. Scale and limits

Current practical scale. The tool is intended for rapid prototyping in the approximate range of 25 to 30 qubits at the upper end, depending on layers, iterations, restarts, memory, runtime limits, and whether exact-state simulation or sampling is used. Smaller examples are better for fast iteration and discussion.

Why the limit matters. Statevector simulation grows exponentially with the number of qubits. Increasing the number of qubits, QAOA layers, iterations, or restarts can materially increase runtime and memory requirements. The online version therefore applies license-level limits and pre-run runtime estimates.

Recommended starting point
Start with a small workbook, validate the model structure, compare classical and QAOA outputs, and only then increase qubits, layers, and iterations. This is usually more useful than starting with the largest possible run.

9. Online and offline usage

Mode	Best suited for	Notes
Online tool	Fast demonstrations, controlled beta testing, and shared access.	Runs on Cloud Run with license keys, runtime limits, and one active run per key.
Offline tool	Local PoC work, sensitive data, longer experiments, or customized notebooks.	Can be configured for more detailed analysis, subject to local hardware and installation.
Hybrid approach	Client workshops and staged PoCs.	Use the online tool to demonstrate the workflow, then move to offline analysis for deeper client-specific work.

10. Good practice for client tests

- Use dummy or non-sensitive data for first tests.
- Keep the number of variable options small until the model setup is validated.
- Review budget and covariance assumptions before interpreting outputs.
- Compare the quantum result with the classical benchmark rather than viewing it in isolation.
- Treat the result as an experimental prototype output, not as portfolio advice.
- Use one license key per organization or tester to keep access, usage, and run limits clean.

11. Suggested first test agenda

1. Ten-minute orientation: problem definition, workbook structure, and fixed versus variable positions.
2. Fifteen-minute setup: upload workbook, inspect model size, and review runtime estimate.
3. Fifteen-minute classical baseline: run classical_only and review result tables.
4. Thirty-minute QAOA run: run qaoa_limited with conservative settings and compare outputs.
5. Fifteen-minute discussion: what worked, what model assumptions matter, and what a next PoC would require.

12. Important disclaimer

QAOA RQP is an educational and prototyping tool. It is intended to make quantum optimization concepts tangible and to support structured exploration of use cases. The outputs are not investment advice, do not replace professional portfolio construction, and should not be used for production decision-making without independent validation.

13. Intellectual property and usage notice