Table of Contents

• QxBranch

• Quantum Applications Strategy

• Semidefinite Programming

• Maximum Risk of a Portfolio

• Quantum Semidefinite Programming
QxBranch Overview

- Founded in 2014
- 25 person team
- HQ in Washington, DC; offices in the UK and Australia
- Partnership-focused engagements

Predictive Analytics
- Pricing and Market Analysis
- Risk
- Behavioral Analytics

Quantum Computing
- Simulation and developer tools
- Application benchmarking
- Algorithm development

QxBranch is a predictive analytics company. At QxBranch, we apply the latest and most advanced data science techniques, to help businesses make more informed and accurate high value decisions, with the ultimate goal of increasing revenue, reducing costs and streamlining operations.

Customers Include
- Four of the top 10 global retail & investment banks
- Top 10 oil & gas company
- Two Top 5 global pharmaceutical companies
- National financial regulator
- Big 4 Australian retail bank
- Asian-Based Hedge Fund with $1B+ AUM

Quantum Computing
QxBranch is a leader in the rapidly emerging and potentially transformative field of Quantum Computing, which is expected to help companies apply exponentially more powerful data science to solve their most complex business challenges—many of which are unimaginable today.
QxBranch operates in the quantum computing software space, providing quantum computing and industry domains knowledge and the tools that allow software engineers to work in familiar environments such as Python and C++.
We help Our Clients begin their quantum computing journey now, to ensure they will be best positioned when quantum computing reaches maturity.

**Application Identification**
- Identify quantum computing applications that align with business needs and assess their near, medium and long term potential impacts.

**Application Design and Build**
- Conduct software development to characterize applications on current hardware and assess the impact of future hardware.

**Application Evaluation**
- Perform fundamental analysis on identified applications to evaluate suitability for proof of concept development.

**Strategic Development**
- Assist in making smart, efficient investments in quantum computing technologies.

**Education and Training**
- Hands on quantum computing experience.
- Insight into future applications.
- Workforce development.
QxBranch has worked with customers on potential impact of Quantum Computing in the financial sector and implemented various prototypes. QxBranch has also developed applications using quantum machine learning and optimisations in other industries. Examples are: seismic image coherency using quantum neural networks with large oil and gas company, resources optimization with global logistic company and marketing optimisation with a media company.

<table>
<thead>
<tr>
<th>Application Areas</th>
<th>Applications</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Machine Learning and Statistical Methods</strong></td>
<td><strong>Feature selection</strong></td>
<td>Quantum-accelerated $L_0$-regularised linear regression algorithm.</td>
</tr>
<tr>
<td></td>
<td><strong>Probabilistic modelling</strong></td>
<td>Modelling of distributions with Hamiltonian: [ H = \sum_{k=1}^{M} \left( df(t + T, y</td>
</tr>
<tr>
<td></td>
<td><strong>Client behaviour analysis</strong></td>
<td>Quantum clustering algorithms for both AQC (Adiabatic Quantum Computing) and UQC (Universal Quantum Computing).</td>
</tr>
<tr>
<td><strong>Optimisations</strong></td>
<td><strong>Beta hedging</strong></td>
<td>Discrete quadratic optimisation with constraints.</td>
</tr>
<tr>
<td></td>
<td><strong>Volatility surface fitting</strong></td>
<td>Hybrid quantum-classical solver that rapidly generates models, evaluates them using digital algorithms and feeds the results back into the quantum generation process.</td>
</tr>
<tr>
<td></td>
<td><strong>Portfolio optimisation</strong></td>
<td>Various, including mixed integer non-linear programming.</td>
</tr>
<tr>
<td></td>
<td><strong>Collateral allocation and multi-stage arbitrage</strong></td>
<td>QAOA (Quantum Approximate Optimization Algorithm) and Quantum Adiabatic Algorithm (QAA) applied to graph theory.</td>
</tr>
<tr>
<td></td>
<td><strong>Cyber security and compliance</strong></td>
<td>Subgraph isomorphisms through quantum optimisation of: [ H = A \sum_{i} \left( 1 - \sum_{v} x_{v,i} \right)^2 + B \left( \sum_{E_L \rightarrow E_2} x_{ui}x_{v,j} + \sum_{-E_L \rightarrow E_2} x_{ui,j}x_{v,j} \right). ]</td>
</tr>
</tbody>
</table>
Going Beyond Point Solutions

• Working with clients on real, but specific, industry problems is paramount, but, without an appropriate framework, it can lead to:
  ▶ Designing quantum solutions with no practical applicability in the near-term
  ▶ Developing point solutions with limited generalisability

• More in general, the problem could be phrased into “how do we fit the work done with clients into a coherent quantum applications strategy?”

• To try to answer, let’s take a step back…
Quantum Algorithms – Present Status

- Exponential speedups, e.g. Shor’s algorithm
- Polynomial speedups, e.g. Grover’s algorithm
- Heuristics, e.g. quantum annealing

- In the NISQ (Noisy Intermediate Scale Quantum) era, boson sampling and random circuits are considered for quantum supremacy
- Can anything useful be built on NISQ computers?
Designing a Quantum Application: Why is it Difficult?

• What constitutes a good quantum application? Consider the example of Shor’s algorithm:

  ▶ **Speedup**: exponential, making full use of the size of the space $2^n$ ($2^n \geq N^2$ is necessary for the converge of the continued fractions expansions)
  ▶ **Classical-quantum interfaces**: extremely simple, the input is a number (although it actually its properties define an implicit database) and the output is also a number
  ▶ **Real-world applications**: encryption

Unfortunately, the algorithm lacks a last important feature: *practical feasibility*

• Finding quantum applications satisfying all requirements has proven to be intrinsically difficult
Searching for More Applications

An operational challenge adds to the inherent difficulty: different highly specialised skills need to be married.

<table>
<thead>
<tr>
<th>Skill Ecosystem</th>
<th>Algorithms and Applications</th>
<th>Examples of Algorithms and Applications Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physicists, mathematicians,...</td>
<td>Class of problems, e.g. optimisations</td>
<td>Does the class of problems correspond to a large class of useful applications?</td>
</tr>
<tr>
<td>QC translators</td>
<td>Fundamental algorithm, e.g. semidefinite programming</td>
<td>Can the algorithm work on NISQ computers? Should algorithms for quantum annealing, or gate based quantum computers, or available in different forms for both, be considered?</td>
</tr>
<tr>
<td>QC platform</td>
<td>Application, e.g. portfolio analysis</td>
<td>What are all the known classical applications and do they have the right features to benefit from the quantum algorithm?</td>
</tr>
<tr>
<td>Quality software engineering</td>
<td>Industry specialisation, e.g. capital markets</td>
<td>Does the application actually correspond to the industry’s real use cases and how much can it be used in other industries?</td>
</tr>
<tr>
<td>Industry experts</td>
<td>Client specialisation, e.g. Bank X parametrisation</td>
<td>How much can Bank X application be reutilised for other banks?</td>
</tr>
</tbody>
</table>
A meaningful quantum applications strategy needs to be founded in a cohesive skill ecosystem, ranging from academic to industry and software engineering skills.

Practically, more than one direction in the Algorithms and Applications schema can be used for quantum applications identification. For example:

- Start from the work done with a client, generalise and then find new specialisations.
- Start from algorithms with several use cases in different industries and possibly apt for the NISQ era and specialise.

Assumptions and simplifications can be made. For example, limit the classes of problems to optimisations and machine learning.
Semidefinite Programming

- Semidefinite programmes (SDPs) are a type of convex optimisation, defined as:

\[
\text{OPT} = \max \quad \text{Tr}(CX)
\]

s.t. \( \text{Tr}(A_jX) \leq b_j \quad \text{for all } j \in [M], \)

\( X \succeq 0, \)

where \([M] \doteq \{1, \ldots, M\}\). The variable \(X\) is a positive semidefinite, Hermitian, \(N \times N\) matrix. The constraints \(A_1, \ldots, A_M\) and \(C\) are \(n \times n\) matrices and \(b_1, \ldots, b_M\) real numbers.

- SDPs are a generalisation of linear programs and have many applications: location problems, scheduling, approximating NP-hard problems (e.g. max-cut), quantum information theory, ...
Consider $N$ assets, with expected return $\mu \in \mathbb{R}^N$ and covariance $\sigma \in \mathbb{R}^{N \times N}$. The portfolio’s weights $\omega_i, i = 1, \ldots, N$ represent the fraction of the total wealth held in asset $i$.

In MPT, the estimates of the assets statistics are assumed exact and several strategies to optimise the portfolio expected return $\mu^T \omega$ or variance $\omega^T \sigma \omega$ or functions of the 2 can be considered.

### Examples of strategies

<table>
<thead>
<tr>
<th>Strategy Objective</th>
<th>Examples of strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimise the portfolio variance (original Markowitz formulation)</td>
<td>Variance</td>
</tr>
<tr>
<td>Minimise the difference of the variance of the portfolio and the variance of a reference index</td>
<td>Tracking error</td>
</tr>
<tr>
<td>Minimise measures associated to the semi-deviation (VaR, CVaR)</td>
<td>Downside risk measures</td>
</tr>
</tbody>
</table>

Various constraints can be added to objective functions, to represent the problem in consideration and constraints and strategies can be interchanged.

### Examples of constraints

<table>
<thead>
<tr>
<th>Description of the Constraint</th>
<th>Examples of constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assets allocation must be exact multiple units of a given lot</td>
<td>Round lots</td>
</tr>
<tr>
<td>Constraint on number of assets</td>
<td>Cardinality</td>
</tr>
<tr>
<td>Assets allocations must be contained in intervals</td>
<td>Quantity</td>
</tr>
<tr>
<td>Certain assets must be included in the portfolio</td>
<td>Pre-assignment</td>
</tr>
<tr>
<td>Constraint on quantity invested in each class of similar assets</td>
<td>Class constraint</td>
</tr>
<tr>
<td>Allow for negative quantities of assets</td>
<td>Short portfolio</td>
</tr>
</tbody>
</table>

The introduction of discrete amounts or quantities in constraints makes the problem classically intractable.
Introducing Uncertainties in MPT: Robust Optimisations

• Assuming the distribution of the asset returns is precisely known is typically an unrealistic assumption. The necessary information might not be complete and estimates are subject to estimation errors and possibly to modeling errors, such as assuming that the distributions are stationary

• Introducing uncertainties, the problem of maximum risk analysis, in the variance formulation, can be expressed as:

\[
\begin{align*}
\text{Maximise} & \quad \omega^T \sigma \omega \\
\text{subject to} & \quad \sigma_{ij}^L \leq \sigma_{ij} \leq \sigma_{ij}^U, \quad i, j = 1, \ldots, N \\
& \quad \sigma \succeq 0,
\end{align*}
\]

where \(\omega\) is fixed and \(\sigma\) is the problem variable

• The problem can be expressed as the original SDP:

\[
\begin{align*}
\text{OPT} = \max & \quad \text{Tr}(|\omega \rangle \langle \omega | \sigma) \\
\text{s.t.} & \quad \text{Tr}((-E_{ij}) \sigma) \leq -\sigma_{ij}^L \\
& \quad \text{Tr}(E_{ij} \sigma) \leq \sigma_{ij}^U \\
& \quad \sigma \succeq 0,
\end{align*}
\]

for all \((i, j) \in [M] = [N] \times [N]\) and with \((E_{ij})_{\alpha \beta} = \delta_{i \alpha} \delta_{j \beta}\)
Variations on the Portfolio Robust Optimisation

• The maximum risk analysis problem can be expressed in the same form with different risk measures, e.g. error tracking, VaR, CVaR (the VaR problem is not convex). Higher moments of the returns distribution can also be included in the SDP

• The robust portfolio design problem extends the analysis problem to the determination of the optimal weights, too:

\[
\min_{\omega \in W} \max_{\sigma \in S} \omega^T \sigma \omega
\]

with \( S = \{ \sigma \in \mathbb{R}^{N \times N}: \sigma \geq 0, \sigma_{ij}^L \leq \sigma_{ij} \leq \sigma_{ij}^U \} \) and \( W = \{ \omega \in \mathbb{R}^N: 1^T \omega = 1, \omega_i \geq \omega_{\text{min}}, \mu^T \omega \geq R_{\text{min}} \} \)

• It can be proven that the mix-max problem is in general equivalent to the max-min problem and can be expressed for the constraints above as an SDP on all variables

• Other methods for the analysis problem with an uncertain returns distribution can be employed. Monte Carlo is a common approach, although not easily adaptable to the design problem
In quantum semidefinite programming (QSDP), the quantum computer is utilised to prepare Gibbs states:

\[ \rho = \frac{e^{-\beta H}}{\text{Tr}(e^{-\beta H})} \]

How is it useful?

By Jaynes principle (max entropy principle), given a quantum state \( \rho \), s.t. \( \text{Tr}(A_l \rho) = c_l \), there is a Gibbs state of the form \( \exp(\Sigma_l \lambda_l A_l) / \text{Tr}(...) \), with real numbers \( \lambda_l \) and the same expectation values as \( \rho \) for all \( A_l \).

The solution \( X \) of the SDP can therefore be expressed in the form*

\[ X = \exp \left( \sum_j \lambda_j A_j + \lambda_0 C \right) \]

and the problem becomes that of finding the right \( \lambda_j \).

*In what follows, for a simpler notation, the normalisation will be omitted.
• There is a quantum algorithm for finding \( \text{OPT} \) running in time

\[
\left( T_{\text{Gibbs}} + M^{1/2} \right) \text{poly}(s, R/\varepsilon),
\]

with \( T_{\text{Gibbs}} \) time to prepare the Gibbs state, \( s \) sparsity of \( A_j \), \( R \) upper bound of \( \text{Tr}(X) \) and \( \varepsilon \) additive error

For \( t = 1, \ldots, \log(N) \), \( (R/\eta \varepsilon)^2 \):

**Quantum Computer**

• Create Gibbs states \( \rho_t = \exp\left( \sum_j v_{j,t} A_j \right) \).

• Search for \( i \) s.t. \( c_{i,t} = \text{Tr}(A_i \rho_t) > b_i + \frac{\varepsilon}{2} \), if none exists output solution.

**Classical Computer**

Compute new \( v_i \), such that \( \rho_{t+1} = \exp(\log(\rho_t) - \varepsilon^2 A_i) \).

• Several methods to create Gibbs states, e.g. quantum Monte Carlo, thermalisation, amplitude amplification (Poulin, Wocjan 2009), Quantum Metropolis (Temme et al 2011), ...

• Brandao and Svore showed the worst case complexity QSDP solvers is \( O\left(N^{1/2} + M^{1/2}\right) \)

• Continued improvements in the QSDP solver:
  - Brandao, Svore 2016
  - Apeldoorn, Gilyen et al 2017
  - Brandao, Kalev et al 2017
  - Apeldoorn, Gilyen 2018
  - \( T_{\text{Gibbs}} M^{1/2} s^2 (R/\varepsilon)^{32} \)
  - \( T_{\text{Gibbs}} M^{1/2} s^2 (R/\varepsilon)^{8} \)
  - \( (T_{\text{Gibbs}} + M^{1/2}) s^2 (R/\varepsilon)^{8} \)
  - \( (T_{\text{Gibbs}} + M^{1/2}) s^2 (R/\varepsilon)^{4} \)
  - Exact formulation: \( (M^{1/2} + \gamma N^{1/2}) s \gamma^{4} \), with \( \gamma = (Rr/\varepsilon), r \) rank of \( A_j \)
QSDP in the NISQ Era

• Brandao with Google’s superconductor quantum computer (August 2018 presentation):
  ▶ Simplified version of the algorithm, with scaling $O(T_{\text{Gibbs}} M/\varepsilon^2)$
  ▶ $C = I, A_i = \sum_{j=1}^{72} \delta_j(i)n_j + \eta_j(i)n_j(n_j - 1) + \sum_{l \sim k} g_{k,l}(i)(a_l^\dagger a_k + a_k^\dagger a_l)$
    (Bose-Hubbard model)

• Might not be a real-world problem, but can it be used to demonstrate a quantum speed-up, i.e. quantum supremacy? (note the practical realisation is not simple)
QSDP for the Maximum Risk of a Portfolio

- For the maximum risk portfolio SDP, $M = N^2$, $s \sim 1$, $r = 1$ and it can be shown that in the best case scenario $R/\varepsilon \sim N/\alpha$, with $\alpha = \varepsilon/OPT$. Using the general QSDP solver of Apeldoorn and Gilyen, scaling is given by:
  \[ N^{5+\frac{1}{2}} (1/\alpha)^{5} \]

- The best, general classical method has a run time of $O(M(M^2 + N^\omega + MNs)\text{polylog}(Rr/\varepsilon))$, with $\omega \in [2, 2.373)$, which for the maximum risk portfolio corresponds to scaling as
  \[ N^6 \text{polylog}(N/\alpha), \]
  which is more efficient than the quantum method, unless very low precision, or a portfolio of unrealistic size, are considered

- However:
  - The QSDP solver has not been specialised for the specific SDP problem. Results have been obtained for specialised solvers, even for graphs problems, which typically have high $R/\varepsilon$ (e.g. Brandao, Franca, Kueng 2018)
  - The general solver’s dependence on $R/\varepsilon$ has improved significantly in the last 18 months and is still being investigated: could there be further improvements? Apeldoorn and Gilyen have shown that the lower bound polynomial dependence of $R/\varepsilon$ is $(R/\varepsilon)^{1/4}$, utilising the $O(N^{1/2}M^{1/2})$ form of the algorithm
  - The requirements in number of qubits is small ($\log(N)$), even for large portfolios. If QSDP quantum supremacy is demonstrated with a prototype and the exponent of $R/\varepsilon$ is reduced, would the maximum risk of a portfolio SDP demonstrate quantum advantage? Implementing the specific $A_i$ on an actual quantum computer, given the present hardware limitations, would also need verification*
Final Considerations

- **Robust portfolios algorithms evolution**: The portfolio design problem with integer quantities is a mixed integer SDP and is NP-hard. It might be possible to design an algorithm combining quantum approximate optimisation and QSDP.

- **Robustness against thermal noise and NISQ**: Is QSDP actually robust against thermal noise? The realisation of Brandao’s prototype could give interesting empirical results.

- **Quantum annealing in QSDP**: Can a quantum annealer be used to prepare the Gibbs states for useful SDPs? Further investigation needed.

- **A final mention**: QSDP, in the quantum input version, produces interesting results for quantum state tomography (shadow tomography).