QUANTUM OPTIMIZATION

Edward Hall
2nd year Physics
Durham

Contents

Overview of Lectures

This section introduces the use of quantum computers in the financial field of optimization. We firstly introduce the concept of **Portfolio Optimization**, which is an especially prominent area in which quantum computers have been tested and may eventually be used on an industrial scale.

We then investigate the three main methods of quantum computation that are used for Portfolio Optimization; we firstly introduce the Hamiltonian concept, which will be central to understand how the methods work. The methods we investigate are **Quantum Annealing**, **Quantum**

Approximate Optimization Algorithm (QAOA), and **Variational Quantum Eigensolver (VQE)**. We will notice how each method bears similarities.

We finish by looking at three applications in which these methods are used in, we investigate their uses in optimizing trading trajectories, arbitrage opportunities, and feature selection in credit scoring.

Definitions

Optimization: The maximization of financial objectives, through allocating resources in the most efficient way.

Portfolio Optimization: The selection of a group of investments (portfolio) that has the highest returns for a given risk.

NP-Hard Problem: A problem can be considered 'NP-hard' if the algorithm used to solve it can be translated into one that can solve any other NP problem.

Hamiltonian: An operator corresponding to the total energy of the system i.e., it's kinetic and potential energies.

Time-Independent Schrödinger Equation: The equation from Quantum Mechanics, which describes the behaviour of a particle in one dimension. The wavefunction is treated to only be a function of position.

Degenerate: In a quantum system, an energy level is described as degenerate if it corresponds to more than one measurable state.

Eigenvalue: A value associated with a specific eigenvector; an eigenvector is a vector that gives a scalar multiple of itself when operated on, by an operator (e.g. The Hamiltonian).

Eigenstate: The state in a quantum system, which corresponds to an eigenvalue of the wave equation

Ground State: The state in a quantum mechanical system at which energy is the lowest possible value. The ground state is nondegenerate.

Adiabatic Quantum Computation: The solving of a financial problem (in our case, portfolio optimization) by modelling a Problem Hamiltonian to a function that is being optimized.

Problem Hamiltonian: The unknown Hamiltonian value used to model the function being optimized.

Initial Hamiltonian: The value for the Hamiltonian chosen initlally for the adiabatic evolution.

Quantum Annealing: The physical implementation of Adiabatic Quantum Computing

Unitary Time Evolution: The time over which the adiabatic evolution occurs, and probability is conserved.

Arbitrage: The making profit through different prices in the same asset in different markets.

Lecture 1

1. Introduction

In finance**, Optimization** is the maximization of financial objectives, through allocating resources in the most efficient way possible. A bank would be inclined to optimize objectives such as net profit and expense reduction.

More generally, optimization can be seen as the process of getting the most out, with the least amount of effort put in; optimization is in fact one of the most fundamental financial problems that needs solving, for companies to operate effectively.

2. Portfolio Optimization

Quantum Computing has recently proven effective in conducting **Portfolio Optimization.** This process selects a portfolio which has the highest returns for a given risk [1]. Significantly, Portfolio Optimization is identified as an **NP-Hard Problem** [2], which is short-hand for a 'Nondeterministic Polynomial Time Problem'. This is a phrase used in computational complexity theory, and we will not go any further than knowing it's definition; a problem can be considered NP-hard if the algorithm used to solve it can be translated into one that can solve any other NP problem [2].

It is extremely difficult, if not impossible, for a classical computer to determine the best portfolio to choose. Thankfully, Quantum Computers can overcome the difficulties that are faced in such NP hard problems. The quantum optimization algorithms can be used on a quantum computer to overcome the difficulty that comes with portfolio optimization. The next lecture explores three methods that have recently been used to solve optimization problems.

[1] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, M.L. de Prado, 'Solving the optimal trading trajectory problem using a quantum annealer', IEEE J, Sel. Top. Signal Process. 10 (6) (2016) 1053–106 https://doi.org/10.1109/JSTSP.2016.2574703.

[2] Woeginger, G.J., 2003. 'Exact Algorithms for NP-Hard Problems: A Survey, in: Advanced Data Mining and Applications. Advanced Data Mining and Applications', pp. 185–207, https://doi:10.1007/3-540-36478-1_17.

Lecture 2

3. The Hamiltonian

Prior to exploring our methods, we are first required to understand the concept of the **Hamiltonian**, denoted $\widehat{\boldsymbol{H}}$. In classical mechanics, the Hamiltonian is a mathematical **operator** that corresponds to the total energy of a system, i.e., the kinetic and potential energy of the system:

$$\widehat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)$$

This also applies to systems at the atomic level that we see in quantum mechanics, which is physics that describes nature at the very small scale; when the Hamiltonian operates on a wavefunction, $\boldsymbol{\psi_n(x)}$, we are led to the famous **Time-independent Schrödinger equation**:

$$\widehat{H}\psi_n(x) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi_n(x) + V(x)\psi_n(x)$$

$$\widehat{H}\psi_n(x) = E\psi_n(x)$$

which describes the behaviour of a particle at the quantum level. It is also important to note that an energy level of a quantum system, n, (which you would have seen in A level physics) is known as being **degenerate**, if there is more than one measurable state that corresponds to it.

Mathematically, this is represented with our Hamiltonian above, and this shows how our system can have more than one state (**eigenstate**) corresponding to the same energy value (or **eigenvalue**). Luckily for us, the only state we will be dealing with is the **ground state**, which is non-degenerate, meaning it only has one possible measurable state.

4. Quantum Annealing

The first method we investigate is Quantum Annealing. Historically, it is the first method out of the three that was used; it also is a change of scene from the 'gate functions' that we have previously seen so far in the course. Interestingly, we will also notice how each method as we go down the list is related to the previous method.

Figure 1. A D-wave machine.[3]

In quantum optimization, the algorithms used involve a method called **adiabatic quantum computation**[4]**.** The most well-known implementation of this is **quantum annealing**.

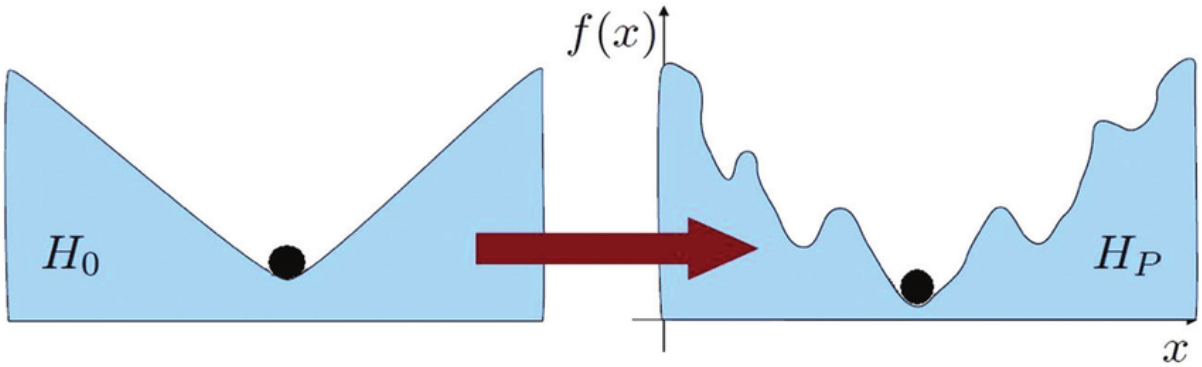

Figure 2. A simplified 2-d graph representing adiabatic evolution.[5]

To describe this, we are effectively modelling the optimization problem to a quantum system that you may have seen at A level. The ground state of a **Problem Hamiltonian**, $\mathbf{H_p}$, represents the cost function we want to minimize through optimization. Finding the ground state of this Hamiltonian is particularly difficult; adiabatic quantum computing overcomes this by starting with an **initial Hamiltonian**, $\mathbf{H_0}$, which we already know how to prepare (as we have all its spins polarized in a particular direction).

During a long period, T, H_0 is **adiabatically deformed** to H_p. This means that over a long adiabatic evolution, while keeping the energy levels of the Hamiltonians separate and other external conditions like temperature constant, it is very probable that we are left in the ground state of the Hamiltonian that corresponds to our solution, H_P.

[3] Paul Rincon, D-Wave: 'Is $15m machine a glimpse of future computing?' (2014), https://www.bbc.co.uk/news/science-environment-27264552
[4] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum Computation by Adiabatic Evolution, (2000). Quant-ph/0001106, http://refhub.elsevier.com/S2405-4283(18)30057-1/sbref0007
[5] Hari Krovi, 'Models of Optical Quantum Computing', (2017), https://www.researchgate.net/publication/315416652_Models_of_optical_quantum_computing

This is known as the 'universal model of quantum computation'[6], which means it effectively can perform any quantum algorithm. Typically, the Hamiltonian values that we start with involve:

$H_0 = \sum_i \sigma_i^x$ – The sum of magnetic fields in the x direction

$H_P = \sum_i h_i \sigma_i^z + \sum_{i,j} \mathcal{J}_{ij} \sigma_i^z \sigma_j^z$ – The first and second terms on the left-hand side are the arbitrary couplings and magnetic fields (in x and z directions) respectively.

Many repeats of the above process are made to obtain many proposals for H_P. After many interpolations, the best solution is chosen by the D-wave machine. This concludes the first method of solving optimization problems with quantum computing.

Quantum annealing is the physical implementation of adiabatic quantum computation. The process has the same objective as classical (or simulated) annealing. Thermal fluctuations cause the system to 'classically' jump between different local minima in the energy landscape (red path). As the temperature lowers, the probability of the system jumping to a worse solution tends to zero. In the case of quantum annealing, the jumps are instead driven by quantum tunnelling events (blue path), rather than a change in temperature.

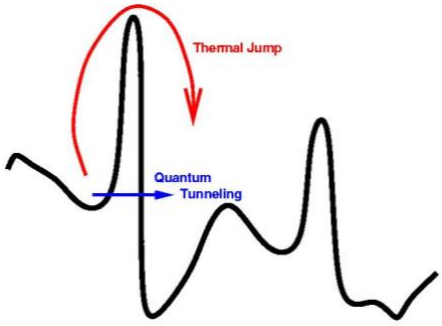

Figure 3: A simplified 2-D graph showing the difference between Classical and Thermal annealing[7].

5. Quantum Approximate Optimization Algorithm: QAOA

Our second method of interest is **Quantum Approximate Optimization Algorithm**, denoted **QAOA**. We will notice that QAOA and Quantum Annealing are very similar (is in fact described as 'Counterdiabatic'); the only difference is that in QAOA, the **unitary time evolution**, denoted U(t), is broken up into discrete steps. Mathematically it is described below:

$$U(t) \rightarrow \prod_{i=1}^{m} e^{-iH_0\alpha_i} e^{-iH_P\beta_i}$$

[6] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, O. Regev, 'Adiabatic quantum computation is equivalent to standard quantum computation', SIAM J. Comput. 37 (2007) 166–194, https://doi.org/10.1137/080734479.

[7] Arnab Das, Quantum Annealing and Other Optimization Methods workshop, (2005), https://www.researchgate.net/publication/252099465_Quantum_Annealing_and_Other_Optimization_Methods

Let's quickly explore what the above 'Pi Notation' means. You will be familiar with sigma notation from A level, where for example:

$$\sum_{i=1}^{5} i = 1 + 2 + 3 + 4 + 5 = 15$$

The notation above is known as 'Pi notation'; similar sort of idea to our sigma notation, the only difference is that the Pi notation (or product notation) multiplies each value together, for example if used pi notation on the above example:

$$\prod_{i=1}^{5} n = (1)(2)(3)(4)(5) = 120$$

In the case of our QAOA equation, we can now recognise the terms on the arguments of our two exponentials (our 'discrete gates'); the *i* values are the inputs to the notation, and we also recognise the H_0 and H_P terms, from our quantum annealing method, as the initially chosen and 'problem' Hamiltonian values respectively.

Notably, we identify α_i and β_i, which are the times over which we run the associated Hamiltonians for, which are left as free parameters over which we optimize. We are therefore able to carry out sampling to find the outcome's expected energy value, which corresponds to the ground state of H_P, our target. To do so, a quantum circuit diagram like that shown below is implemented:

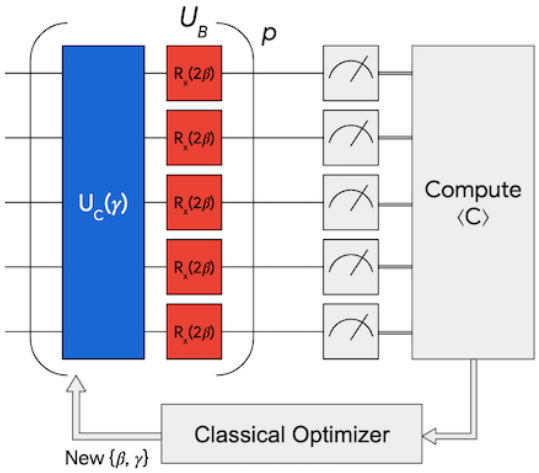

Figure 4: Quantum Circuit Diagram for QAOA [8]

Choosing a certain value for m, we can optimize the time parameters α_i and β_i for each Hamiltonian, until we converge to our solution for H_P. A value for m (i.e the number of iterations) is chosen, and the energy measured from the quantum circuit's output is used to optimize the free parameters that are the alphas and betas. This process is repeated until there is a convergence to H_P.

[8] Billy Lamberta, Ryan LaRose, Mathew Harrigan (2021), QAOA example problems, https://github.com/quantumlib/ReCirq/blob/master/docs/qaoa/example_problems.ipynb

6. VQE: (Variational Quantum Eigensolver)

Just as QAOA has similarities to Quantum Annealing, we notice that our final method, **Variational Quantum Eigensolver**, has similarities to QAOA; The ideas and processes are effectively the same between the two methods, the only difference is that the parameters in our implemented circuit used is chosen by us, whereas in a QAOA circuit, the variable remains as the times over which the Hamiltonians are run for.

An example of a circuit used in VQE involves the set of one qubit rotations, $U_{N,0}$, which passes through a fixed entangling gate U_{ent} a certain number of times. The parameter used in this case is the actual angle of the one qubit rotations, θ:

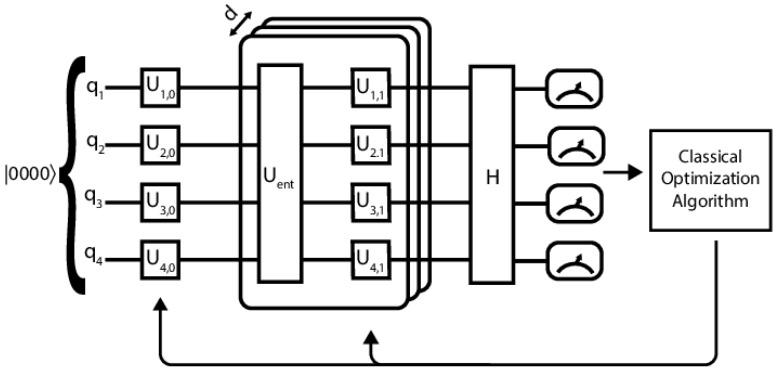

Figure 5: Quantum circuit diagram for VQE [9]

An initial value of the angle is chosen, and having run the circuit, the energy from the outcome is sampled. The sampling can then be used to choose the modification of the parameters to go toward the ground state. Analogously to QAOA, we will reach the solution for the problem Hamiltonian H_P, after several iterations.

We now have covered the methods that can be used to explore three prominent applications of quantum computers in portfolio optimization.

[9] R. J. P. T. de Keijzer, Victor Colussi, B. Škorić, Servaas Kokkelmans, (2021), Optimization of the Variational Quantum Eigensolver for Quantum Chemistry Applications, https://www.researchgate.net/publication/349025671_Optimization_of_the_Variational_Quantum_Eigensolver_for_Quantum_Chemistry_Applications

Lecture 3

7. Optimal Trading Trajectory

The process of finding an **Optimal Trading Trajectory** is more famously known as **'Dynamic Portfolio Optimization'**. The aim of this application is, as it says in the name, to find the optimal trajectory in the portfolio space, while considering transaction costs and market impact.

Essentially, over a long period of time, the optimal composition of a portfolio is found every day until the portfolio's returns are maximised. If possible, we also wish to minimize the risk. In this case, the cost function that we are optimizing is the return, denoted ω and is shown mathematically below:

$$H_P = \omega = \sum_{t=1}^{T} (\mu_t^T \omega_t - \frac{\gamma}{2} \omega_t^T \sum_t \omega_t - \Delta\omega_t^T \Lambda_t \Delta\omega_t + \Delta\omega_t^T \Lambda_t' \omega_t)$$

Terms:

- μ_t^T: Forecasted returns
- ω_t: Holdings at time t (these terms come together to form the variable that we are trying to optimize).
- γ: Risk aversion
- $\sum_t$: Forecast covariance tensor
- $(\Delta\omega_t^T \Lambda_t \Delta\omega_t + \Delta\omega_t^T \Lambda_t' \omega_t)$: Transaction cost.
    - This is the fee paid to the bank when buying or selling. This effectively is the term that makes the cost function the most realistic.

Given we know each of these terms, our cost function, ω, can be mapped to a problem Hamiltonian H_P, and processed using one of the above methods such as a Variational Quantum Eigen solver.

Graphically, we represent what we have been discussing; the efficient frontier curve shown shows the set of portfolios that offer the highest expected return for a certain level of risk, or alternatively, the lowest risk for a certain level of expected return.

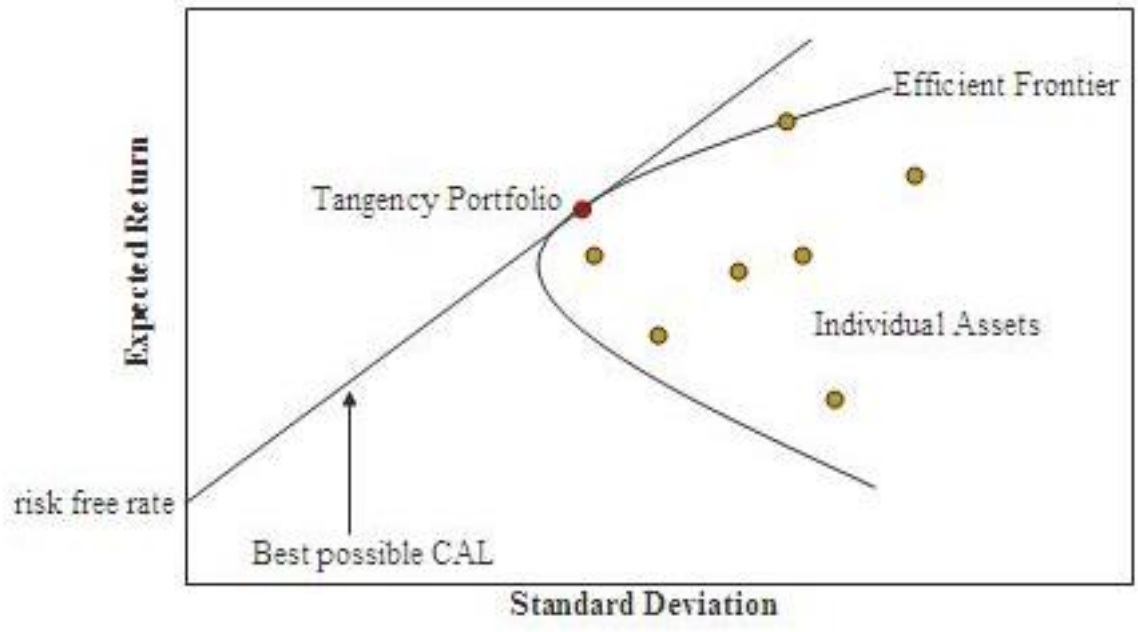

Figure 6: Graphical representation of Optimal Trading Trajectory[10].

The Optimal Portfolio (or 'Tangency Portfolio') on the graph, is the portfolio on this frontier with the highest return over risk.

8. Optimal arbitrage opportunities

Arbitrage is defined as making profit through different prices in the same asset in different markets, making this application especially useful in currency trading (you may also know this as **FOREX trading**). For example, this could be changing sterling to dollars, then later changing dollars back to sterling, and making a profit in the small process. This is known as cross-currency arbitrage.

The goal of finding the optimal arbitrage opportunity is another NP-Hard problem; it is to identify cycles like these which provide a positive return. In our case, optimal arbitrage opportunities can be identified by a quantum annealer. Starting with a directed graph:

- The nodes, i, represent the assets we are interested in.
- The directed edges are weighted with the conversion rate c_{ij} (note that $c_{ij} \neq c_{ji}$).

We are now able to identify the most profitable cycle in this directed graph; using the Boolean variables; x_{ij} (i.e., the binary approach of 1 or 0) is 1 if ij belongs to the profitable cycle, and 0 otherwise. We can represent this as:

$$\omega = \sum_{(i,j)\in E} x_{ij} \log(c_{ij}) - M_1 \sum_{i \in V} \left(\sum_{j,(i,j)\in E} x_{ij} - \sum_{j,(j,i)\in E} x_{ji} \right)^2 - M_2 \sum_{i \in V} \sum_{j,(i,j) \in E} x_{ij} \left(\sum_{j,(i,j) \in E} x_{ij} - 1 \right)$$

[10] Sanket Karve, (2020), Automating Portfolio Optimization and Allocation using Python, https://towardsdatascience.com/automating-portfolio-optimization-using-python-9f344b9380b9

Terms:

- E is the edges of the graph
- V is the vertices of the graph
- 1st term: $\sum_{(i,j)\in E} x_{ij} \log(c_{ij})$ - logarithm of the cycle's cost
- 2nd term: $M_1 \sum_{i \in V} \left(\sum_{j,(i,j)\in E} x_{ij} - \sum_{j,(j,i)\in E} x_{ji}\right)^2$ – a flow constraint.
- 3rd term: $M_2 \sum_{i \in V} \sum_{j,(i,j) \in E} x_{ij} \left(\sum_{j,(i,j) \in E} x_{ij} - 1\right)$ – ensures that cycles can go only once through an asset.
- M_1 and M_2 are penalty parameters.

The problem has become a quadratic unconstrained binary optimization problem, also known as QUBO. Significantly, this means it can be used with a quantum annealer.

9. Optimal feature selection in credit scoring

Optimal feature selection in credit scoring is a problem that can be solved using either Optimization or machine learning (which Jack will cover in your next set of lectures). Credit scoring is the identifying of the level of risk that is associated with a loan, by considering factors like the borrower's income, age, etc. This enables banks to identify the borrower as a high or low risk customer.

The focus is on the data of past applicants; the 'creditworthiness' (i.e., their worthiness for a loan) of a potential borrower can be determined from this data. The related problem is on the selection of the data of past applicants, to be used to determine the creditworthiness of new applicants; for example, some of the data may be irrelevant, or we may not even have access to all the data. We require two parameters to represent this mathematically:

- We define a Matrix, U, which is *i* rows by *j* columns (i.e. i*j entries)
  - The row, *i*, represents numerical values
  - The column, *j*, represent the applicant's factors, such as age.
- We define a Vector, V, which shows the record of past credit decisions.

Putting this together:

$$\omega = -\left(\alpha \sum_{j=1}^{n} x_j \left|\rho \vee_j\right| - (1 - \alpha) \sum_{j=1}^{n} \sum_{k \neq j}^{n} x_j x_k \left|\rho_{jk}\right| \right)$$

Terms:

- α is the control parameter which controls the relative wright between the two terms.
- x_j is a binary Boolean variable, which means it equals 1 if *j* is in the subset of "selected" features, and 0 otherwise.
- ρ is the correlation between column j of U and V (influence on outcome).
- ρ_{jk} is the correlation between columns, j, k of U (known as mutual dependence).

Effectively, the first term corresponds to the influence that the application's factors have in the credit outcome, and the second term models the independence between the features. In this form, ω can be optimized using a quantum annealer.

10. Conclusions

We have now seen three prominent methods of quantum computing in the financial field of portfolio optimization, and we have also gone onto explore the application of these methods in financial markets; primarily through the modelling of a Problem Hamiltonian to the cost functions that we are wishing to optimize.

Although the previously discussed methods in this lecture are very much in the early stages of their use in the commercial world [11], it is in my opinion exciting that extremely difficult problems in a variety of industries, like that of portfolio optimization in finance, will be within reach of financial companies across the world.

It therefore comes as no surprise that the patents in quantum technology since 2012 have skyrocketed [12];

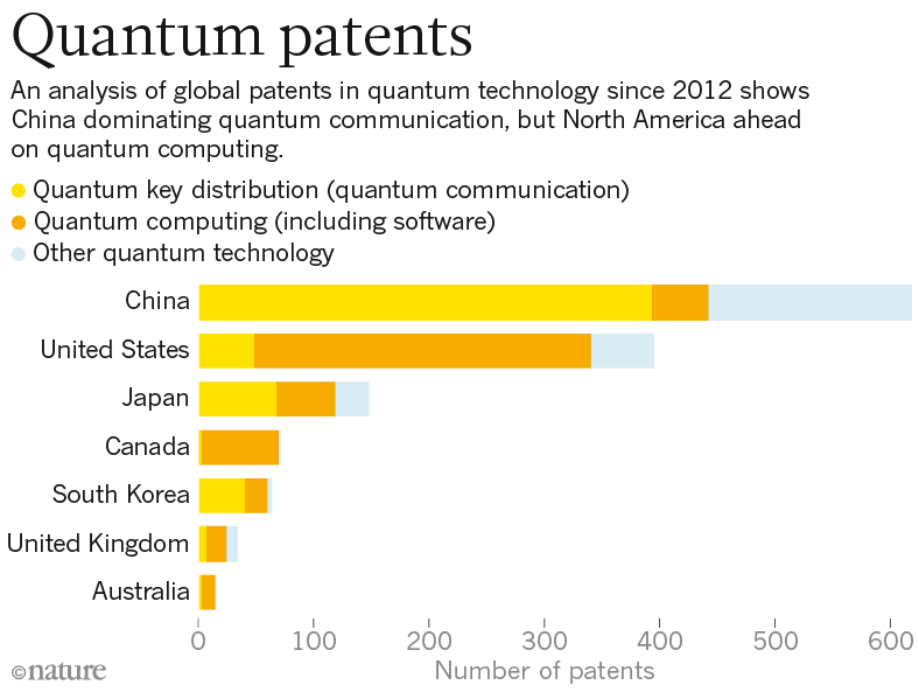

Figure 7: Graph to show increase in global patents in quantum technology since 2012 [12].

which shows the potential for quantum technologies, not only in finance, but across all industries during our lifetimes.

[11] A. Milne, M. Rounds, P. Goddard, Optimal feature selection in credit scoring and classification using a quantum annealer (2017) https://1qbit.com/whitepaper/optimal-feature-selection-in-credit-scoring-classification-using-quantum-annealer/

[12] Elizabeth Gibney, Quantum gold rush: the private funding pouring into quantum start-ups (2019) https://www.nature.com/articles/d41586-019-02935-4