

Introduction

Someday, quantum computers will be powerful enough to provide elegant solutions to combinatorial optimisation tasks such as the graph colouring problem. However, it will take a couple of years before we enter the age of fault tolerant quantum computation. At present we are in the NISQ (noisy intermediate scale quantum) era. This means the present quantum processors have high error rates, poor hardware capabilities and a limited 'quantumness' of the qubits (the quantum version of a bit).

Working with NISQ machines has provided great insights into how quantum algorithms operate. It led to the emergence of a powerful class of algorithms called variational quantum algorithms (VQEs) which rely on the principle of minimising the energy of a quantum system. Researchers have shown that hybrid classical-quantum algorithms such as the QAOA (Quantum Approximate Optimization Algorithm) can give a good approximation to the graph k-colouring (Max-K-Cut) problem. One of the benefits of QAOA is that it works well with NISQ machines. As stated by Phd student Fergus Barret working on quantum simulations at UCL "QAOA is provably good as compared to other variational quantum algorithms which just use a ... anstz." (CHECK)

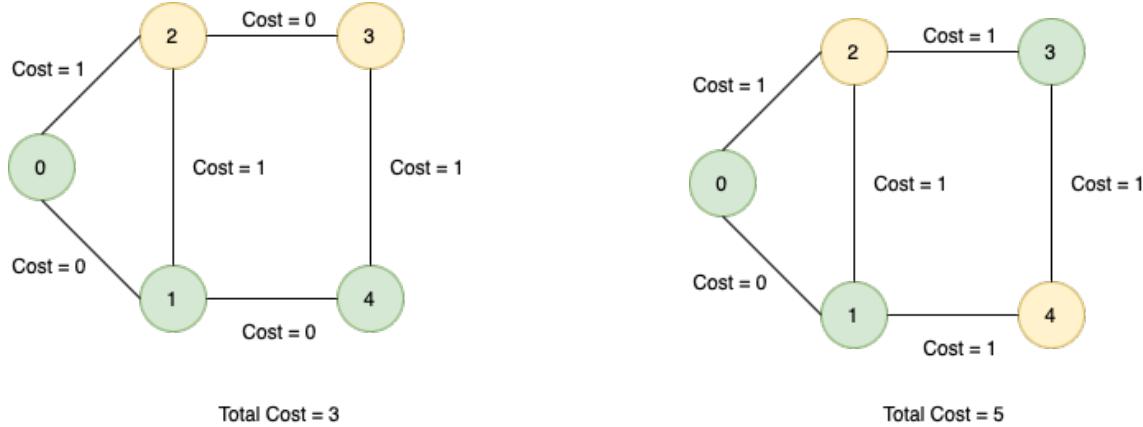
The Max-k-Cut problem has been around long before quantum computers existed. Computer scientists classify it as an 'NP hard' problem. This basically means there is no algorithm that can solve it in polynomial time. There are however several classical algorithms that can find approximate solutions to Max-k-Cut, just like the QAOA. Just because an algorithm is quantum doesn't directly mean that it is better than its classical counterparts. New research has come out which has shown that the classical Newman algorithm outperforms the QAOA. However the same paper also discusses a recursive modification to QAOA, the RQAOA, which has proven to be better than Newman in solving Max-K-Cut.

So, what is Max-k-Cut exactly?

A Max-K-Cut problem is an optimisation of the approximate graph k-colouring problem. It tries to assign K colours to the nodes on a graph. The main aim is to maximise the number of edges whose nodes have different colours. A 'cost' can be defined such that if there is an edge between a pair of nodes and the nodes have been assigned:

1. The same colour, then the cost for this edge is 0.
2. Different colours, then the cost for this edge is 1.

The cost function is simply the sum of all the costs. The problem then boils down to finding a colour combination such that this cost function is maximised.



(a) A terrible assignment of colours leads to a total cost of 3.

(b) A good assignment of colours leads to a total cost of 5.

Figure 1: Solving the Max-2-Cut problem using 2 colours yellow and green. In 1a, the cost between nodes (2, 3) and (1, 4) is zero as they have the same colour. In 1b, swapping the colouring of nodes 3 and 4 leads to the maximum cost of 5 which is the most optimal colouring.

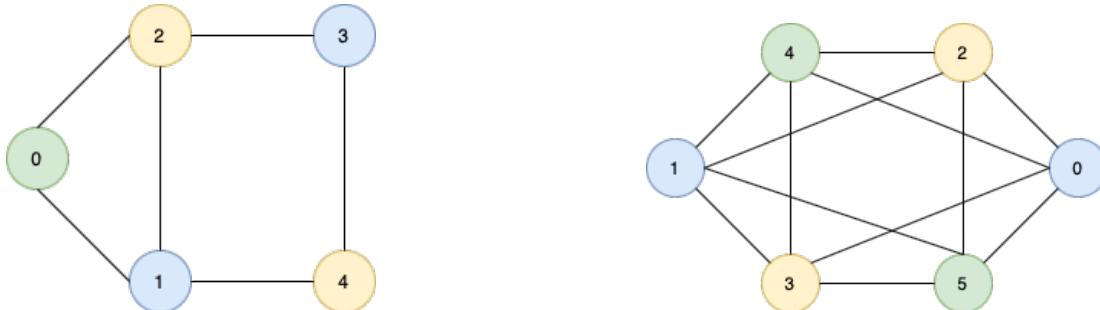


Figure 2: A solution to the Max-3-Cut problem on different graphs. As we can see the endpoints on each edge have different colours. This particular colouring scheme also gives us the maximum cost.

The Old Ways

Before diving too deeply into just how it is that quantum computers are supposed to fit into all this, it is worth taking a moment to look at how this problem has been approached long before qubits got involved.

Graph theory (the mathematical study of graphs, unsurprisingly) can be traced all the way back to 1736 when scientific and mathematical great Leonhard Euler first posed the problem of “The Seven Bridges of Konigsberg”. The task was, quite simply, to find a route through the city of Konigsberg, such that each of its seven bridges would be crossed exactly once. In proving that the problem had no solution, he introduced for the first time the concepts of vertices and edges. And hence, graph theory was born.

Euler’s early work has been expanded upon all the way up to the present day. As things stand, there are two key (classical) approaches that stand out for solving k -colouring problems: the Goemans-Williamson and Newman algorithms.

Due to the difficulty in finding the most optimal colouring, many algorithms instead take an approximative approach. Rather than aiming to find the ideal assigning of colours, the goal is instead a trade-off of time, measured in computational complexity, against accuracy, classified by the “approximation ratio”. Both Goemans-Williamson and Newman take just this approach.

In 1994, Goemans and Williamson proposed a solution to 2 colouring (here k is 2, hence the problem is only to divide the graph into two interconnected sets of colours) that reported accuracies well beyond what had been previously achieved. Their method was, in short, to approximate the complex underlying mathematical equation into a simpler one, known as a semidefinite program which had well known, efficient methods for solving. The trade-off of this approach being that the resulting solutions are not perfect, but are within some fixed margin of the correct answer.

With this significant breakthrough, work then began on taking the Goemans-Williamson algorithm and expanding it to the more general problem of assigning k colours. Despite early extensions by Goemans and Williamson themselves to 3 colours, generalising these ideas effectively for anything above 4 proved difficult. This was until 2018, when Alantha Newman introduced her extended approach, which in the 3 colour case, proved as accurate as Goemans-Williamson. As such, it is the Newman algorithm that all quantum approaches to k -colouring must stand up against.

As mentioned previously, current capabilities of quantum computers are significantly limited. Even so, there is positivity amongst the research community as to what will soon be achieved.

“It is still a very good time for this development and quantum algorithms development in general. The reason is that it is not only the capabilities of quantum computers that determine the design of algorithms, but also the design of algorithms affecting the quantum computers that are built,” said Abhishek Agarwal, a researcher in the quantum division of the National Physical Laboratory.

This power has the potential for great amplification through quantum-classical hybrids known as “variational quantum algorithms”. These work by delegating the roles within an optimisation task to the most suited platform; this usually means a quantum circuit to evaluate some form of cost function, and a classical processor to choose the optimal parameters based on this quantum-calculated result.

These algorithms are an area of active research and have been used to solve very difficult problems such as finding the lowest energy state of quantum systems. However, they do present challenges such as noise from the quantum section making the “training” process significantly harder for the classical optimiser.

Going quantum - The QAOA

The QAOA is one such VQE which was developed in 2014 by Sam Gutmann. It uses a ‘quantum’ version of the cost function, now called the cost hamiltonian, which acts on a quantum state. The quantum state is physically described by the collective state of the qudits (d dimensional quantum bits) whose energy needs to be minimised.

As an example consider the graph shown below where we want to solve the Max-3-Cut problem.

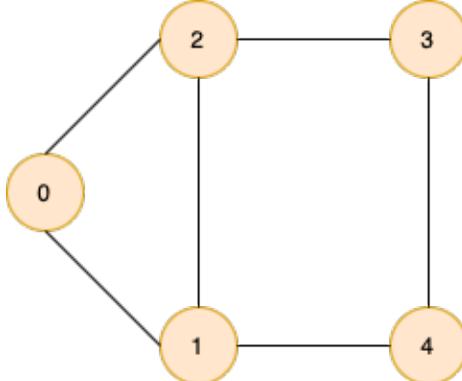


Figure 3: A graph with 5 nodes and 6 edges

In this case each node of the graph will be described using a qutrit (a quantum system with three energy states), with the colour of the node being given by its quantum state. The quantum state of the whole graph can be expressed as a state vector in the computational basis given by $|0\rangle$, $|1\rangle$ or $|2\rangle$ per qutrit.

The algorithm then works as follows :

1) One starts with the initial state being chosen to be a superposition of the computational basis states of the qudit. Using a superposition essentially involves looking at all possible solutions at once, which is the main idea behind quantum computing. For the example graph this state is represented by the state $|+\rangle$ shown in the figure below.

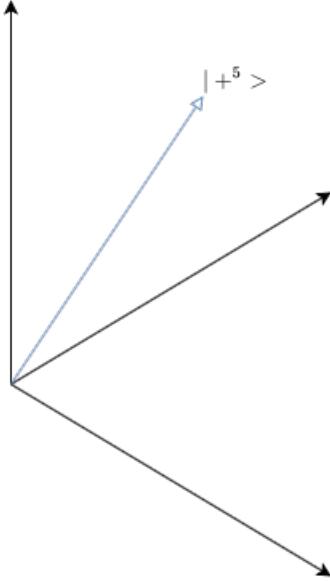


Figure 4: The initial superposition state used is the starting state of any quantum algorithm. The 'space' in which this state exists is called a Hilbert space,

- 2) A unitary operator parameterized by γ and β is applied to this initial state p number of times to evolve it in accordance with the optimization problem that is meant to be solved. γ encodes the cost Hamiltonian while beta represents the mixing dynamics of the state. The parameter p decides the circuit depth.
- 3) Maximising the cost function to get the optimal solution is then analogous to finding the maximum of the expectation value of the cost Hamiltonian for a given quantum state. Finding the optimal γ and β is done classically, via gradient descent or grid search, thus making the QAOA a hybrid algorithm.

For the particular example, two cases of different γ and β used in the evolution of the quantum state are shown. The one on the left evolves the state in the plane of the $|00000\rangle$ and $|11111\rangle$ subspace with an expectation value of 0. The one on the right evolves it into the subspace of $|12020\rangle$ and $|02121\rangle$ with an expectation value of 5.

- 4) Once the optimal γ and β are found, the quantum state is measured in the computational basis. The state each qudit is in ultimately represents the colour of the vertex represented by that qudit thus giving us the solution to our colouring problem.

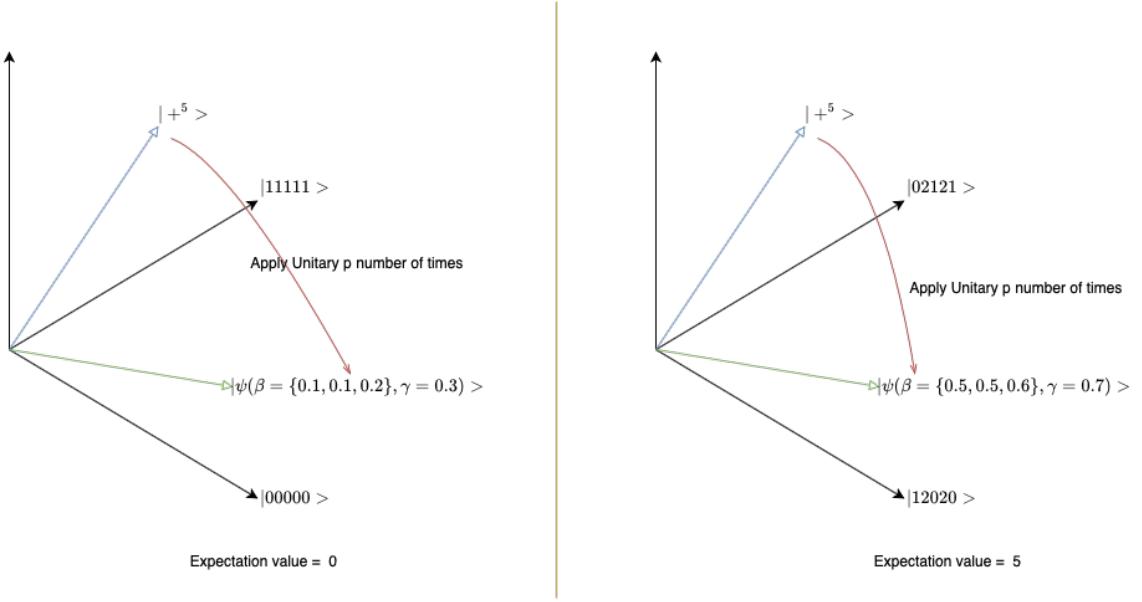


Figure 5: Two possible evolutions of the quantum state using different values of γ and β . γ and β are optimised classically to obtain the maximum expectation value of the cost. Hence, the state on the right is the 'solution' state.

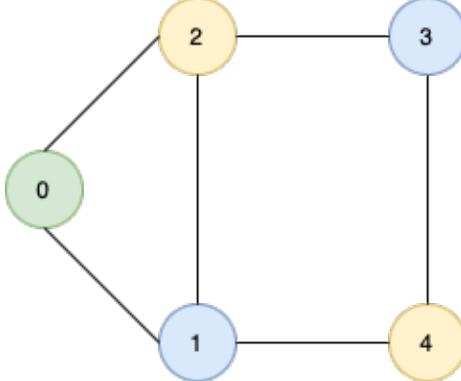


Figure 6: The final colouring scheme as determined using QAOA. The optimal state is the one with the expectation value of 5. Therefore measuring this gives an output state $|02121\rangle$. Colours are assigned for each state (0 is green, 1 is yellow and 2 is blue) and the final solution as shown.

One of the main limitations of QAOA is that it is what physicists call a 'local' algorithm. This means that in order to find the optimal colouring for a graph QAOA needs to see the entire graph. This leads to a large solution space thus reducing the algorithms efficiency of searching for a good approximate solution. Also, it has been shown that in order for the QAOA to outperform the classical algorithms, the param-

eter p needs to logarithmically grow with the size of the graph. Due to the limitation of available hardware, QAOA can only work on small and medium sized graphs. In 2019, authors Sergey Bravi et al. came up with the RQAOA algorithm to tackle these limitations.

QAOA's stronger cousin - The RQAOA

The paper suggests a slight modification to the QAOA, called the recursive QAOA (RQAOA), in order to overcome these limitations. The central idea behind this approach is to recursively reduce the size of the graph for which we need to find the optimal solution. As Phd researched David ... says "RQAOA works by effectively removing those parts of the system that are insignificant or very small (something like that)".

(CHECK) Link non-local

The steps in the RQAOA algorithm, which works for all $K \geq 2$ of the Max-K-cut problem, are described below :

To aid us in our explanation, the same example as above (figure 3) is used to solve the Max-3-cut problem. The colours Green, Yellow and Blue are used to segregate the nodes into their respective sets.

1.) The QAOA algorithm is used to find the optimal parameters β and γ that gives us the maximum expectation value. These are used to generate the optimal state.

2). Using this optimal state, we calculate the correlations between all pairs of nodes that have an edge between them for all possible colours as shown in Figure 8.

3). The pair of nodes with a high correlation are condensed into a single node, while the rest are left alone. This is because the ones that have a high correlation magnitude are related to each other such that if we were to assign a colour to one of the nodes, the colour of the other node can be determined by a constraint. For example, if one node from a pair is coloured blue and the constraint states the difference between the pair of nodes is blue (assigned a number 2), then the colour of the other node can be found by starting at blue and traversing the constraint graph (fig 7) by 2 edges.

Therefore the size of the graph is reduced which means the

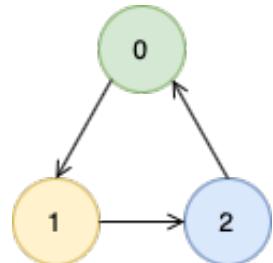


Figure 7: The constraint relation between colours yellow, green and blue.

number of nodes the cost function is dependent on is reduced as well.

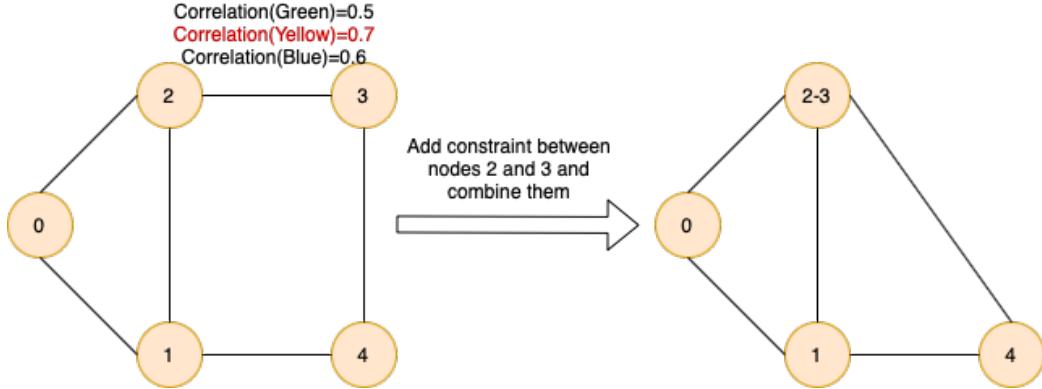


Figure 8: Only the correlations between nodes 2 and 3 are shown. Nodes 2 and 3 and the colour Yellow have the highest correlation magnitude of 0.7. They are condensed into one node and a constraint is added such that the difference in colours between nodes 2 and 3 is Yellow. The new graph now only has 4 nodes.

4). This process is repeated until the number of nodes in the graph meet a predefined cutoff point. Doing this reduces the size of the search space required to span the solution.

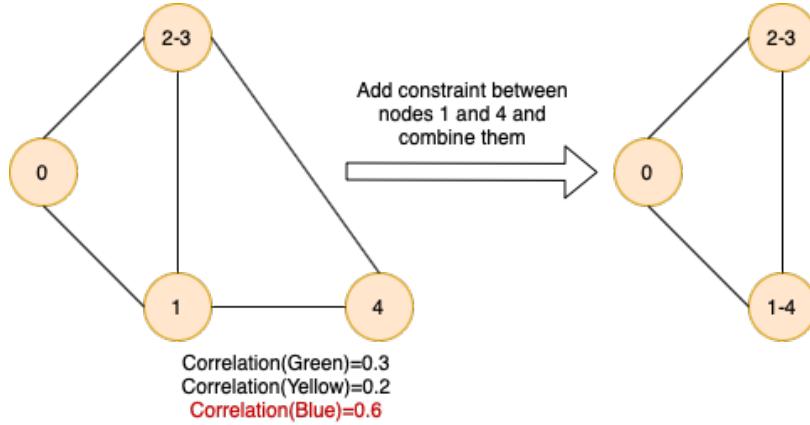


Figure 9: Repeating the process of applying QAOA and recalculating the correlations, we find that nodes 1 and 4 and the colour Blue have the highest correlation magnitude of 0.6. So they are condensed into one node and a constraint is added between them to work out the colouring. The new graph only has 3 nodes. The process stops here as the cutoff point was set to 3 nodes.

5). As the graph is now quite small, an optimal colouring to the nodes can simply be assigned via a brute-force algorithm.

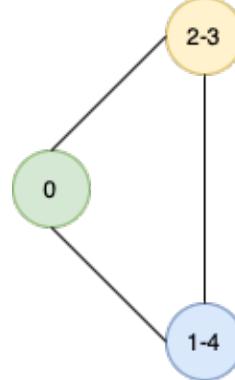


Figure 10: The 3 colours are assigned such that node 0 is green, node 2-3 is yellow and node 1-4 is blue.

6). The final step involves going through all the nodes that were removed in reverse order and applying the constraints to add the colouring.

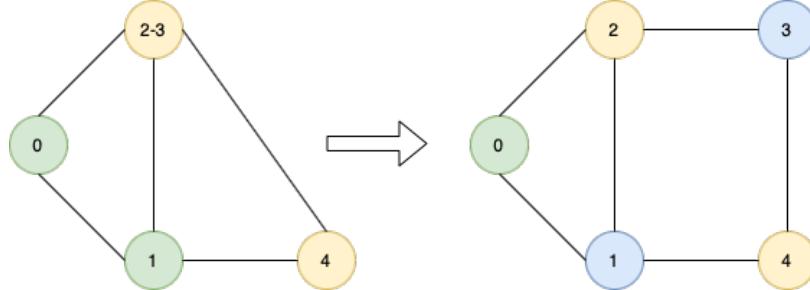


Figure 11: Node 4 is added back and the constraint is applied from node 1, which ultimately results in node 4 having the colour yellow. Similarly adding back node 3 and applying the constraint results in node 3 being coloured blue. All the nodes are now optimally coloured.

- In RQAOA the correlation between the nodes needs to be calculated on every pair making it computationally heavier than QAOA

Simulating QAOA & RQAOA Classically

Whilst the above algorithms may display great promise on a theoretical level for application to Max-k-Cut problems, this means little, in the present era, if there is

not a convenient way to simulate them on classical hardware. Since most researchers have neither the time nor the access to try out new algorithms on actual quantum computers, it is imperative that there is a means of translating them into a classical setting.

In the specific case of QAOA and RQAOA, finding the expectation values of the various states is where this issue comes to a head. Fortunately, a polynomial time method exists for doing just this. In fact, the same process for finding these same expectation values can be adapted to finding the value of the “slope” for each parameter. This “slope” is a measure of in which direction (and by how much) the angles would need to be adjusted for the algorithm to reach the optimal solution.

This determination of this “slope” and the corresponding adjustment for each one of a set of parameters is the underpinning of both machine learning, and many quantum algorithms.

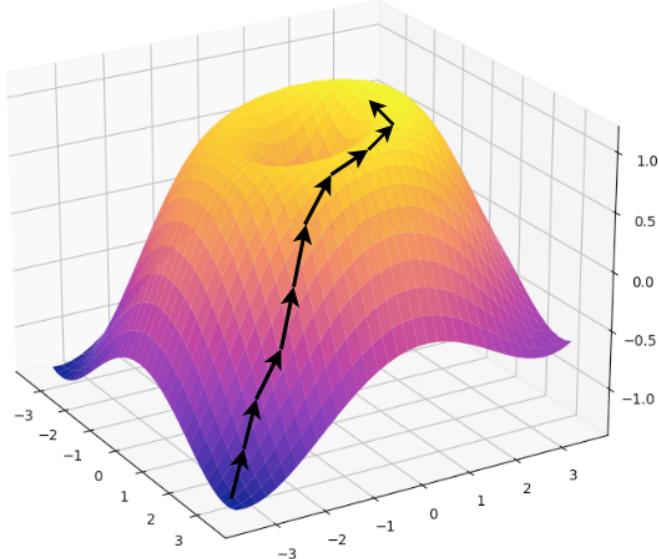


Figure 12: Visualisation of gradient ascent.

However, there is a caveat here that applies to many problems in the domain of quantum optimisation. This caveat is known as “barren plateaus” in the parameter landscape. This means in practice that the various “slopes” for each parameter become small enough to make traversing this landscape (and hence finding the optimal solution) impossible. One could think of this as trying to find the highest point of a completely flat field; much more difficult than identifying the peak in a mountain range.

Fortunately, there is a way round this issue by utilising a machine learning technique called grid search. This method simply involves running the experiment for a fixed set of values for each parameter at constant intervals (hence grid search), and determining which ones performed best. Once a well performing region has been identified, a second search can be performed around that area to further hone the result.

In this specific case, it was possible to perform a grid search over just the value of gamma, and identify the optimal values of the remaining beta parameters via solving analytically. This analytical portion of the optimisation centres around the solution of a degree-4 polynomial - an equation where the largest term is to the 4th power.

This approach, however, only works in the case that the circuit is sufficiently "shallow" (there is not too many layers of quantum gates). Since most recent experiments surrounding QAOA and RQAOA have only used the shallowest configuration possible, this as not been an issue.

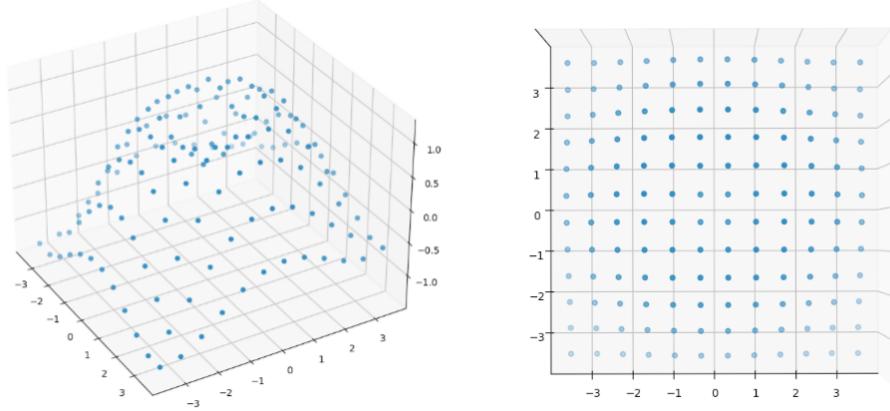


Figure 13: Visualisation of how grid searching works. Left: Perspective view. Right: Plan view.

Grid search can quickly become a costly operation if the number of parameters is large, as the number of configurations to try increases exponentially. As such, this ability to solve for the remaining 3 parameters, without having to repeat the experiment, brings the (classical) computational requirements well into what is feasible on non-specialist hardware.

Putting all this together, researchers were able to run experiments testing just how effective these quantum algorithms are when pitted against the (known to be strong) classical Newman algorithm.

What's the outcome here?

A quantity called an approximation ratio is often used to determine how well the approximate solution determined by approximation algorithms fares with the actual solution. To do so the algorithms are used on problems whose answer is already known.

In this paper, the researchers focus on a particular subset of graphs called d-regular k-coloured graphs. The nodes in the graph are split into k sets where each node in the graph has degree d and the vertices that are in the same set have the same colour and are bipartite i.e they do not have any edges between them as shown in right hand side of Figure 2.

The researchers decided to compare the two quantum algorithms with the classical randomised Newman algorithm. The results were generated by running 100 trials on four different graph sizes and connectivities. What they found was that :

1. in all instances the RQAOA significantly outperforms QAOA for this particular subset of graphs.
2. as the connectivity of the graph is increased, the approximation ratio of QAOA deteriorated significantly, however the RQAOA was less susceptible to this.
3. as the Newman algorithm is randomised, an increase in the connectivity of the graph leads an increase in the variance of the Newman algorithm. In this case the Newman algorithm gave a better approximation ratio than the RQAOA. However the RQAOA outperformed Newman in the average approximation ratio over the 100 trials.
4. in the cases with low graph connectivity the RQAOA outperformed Newman in almost all instances.

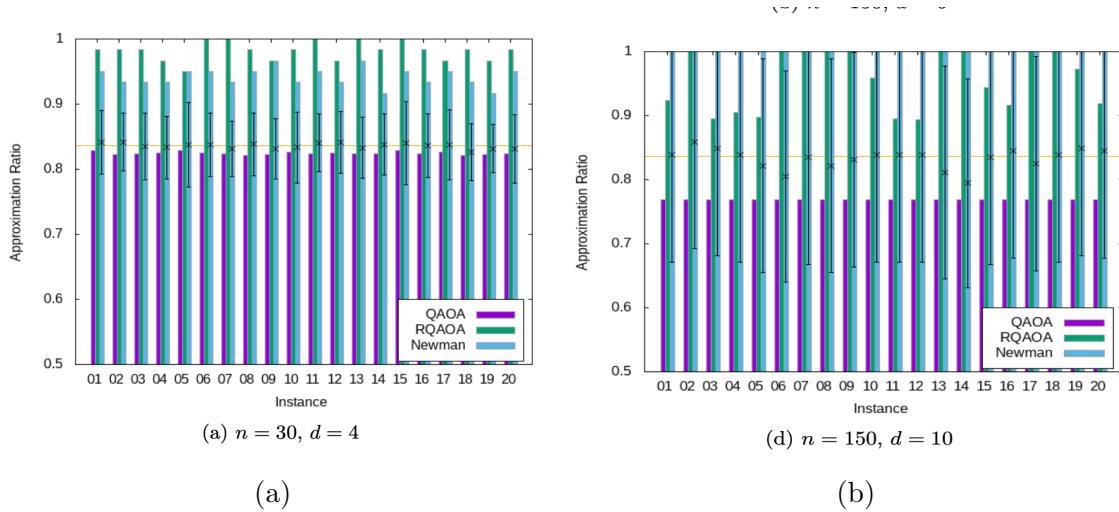


Figure 14: Approximation ratios for QAOA, RQAOA and Newman on two different cases.

As a Phd researcher Sami B... puts it "The results are promising, however I would like to see how RQAOA performs on more relevant tasks like portfolio optimisation etc."

So, what's next?

Finding good approximations to the Max-K-Cut problem will have a huge impact on various fields of physics and computer like timetable scheduling and network and chip design. Also, the Max-k-cut acts as a good toy model to test such approximate algorithms before applying them to real world optimization cases. The next step for these quantum algorithms can be summarised :

1. These algorithms have been evaluated only on a specific subset of graphs. Further evaluations need to be done on a larger set of graphs in order to conclusively state that RQAOA is a superior approach to solve graph colouring.
 2. Further research needs to be conducted to see if RQAOA can be generalised for other combinatorial problems. (CHECK - Davids quote) - use on unsolved cases - use on physically relevant problems.
 3. As stated above, QAOA and RQAOA can be simulated classically with $p = 1$. As QCs get better both QAOA and RQAOA should be tested for $p > 1$ to see how well they perform with respect to classical algorithms.

- use on actual quantum computers. right now efficiency determined by running on classical computers. On QCs need to generate multiple samples to get the expectation value.

After the major setback QAOA faced after A Newman published her classical algorithm, RQAOA offers hope a possibility of a quantum advantage. So it seems that "quantum is awesome" after all.