# NUMERICAL PHYSICS PROJECT REPORT

Martin Plazanet Ghada BEN SLAMA

# CONTENTS

# 1 INTRODUCTION

Optimization problems hold significant importance across a myriad of domains. From logistical challenges to resource allocation and quantum chemistry, the need to efficiently solve optimization problems is paramount. Traditional methods for tackling these problems may face computational bottlenecks as the size of the problem increases. This has led to a surge in interest in quantum computing as a potential game-changer for optimization tasks.

In this project, we focus on a specific quantum variational algorithm : the Quantum Approximate Optimization Algorithm (QAOA). Though this algorithm has been of considerable interest in the past years as an alternative to the Quantum Phase Estimation (QPE) algorithm to solve optimization problems on Noisy Intermediate Scale Quantum (NISQ) devices (the QPE algorthm requiring a completely fault-tolerant quantum computer to yield suitable results), with a potential speed up compared to classical computations. However, such speed up has yet to be demonstrated, and the algorithm has limitations as to its usefullness on noisy devices. Our objective is to explore slightly different yet simple approaches to show how the algorithm might benefit from further improvements. We do not pretend our approaches to be better than the QAOA, but rather aim at highlighting its flaws. As a case study, we focused on solving the well known Travelling Salesman Problem on small graphs.

# 2 THE QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

## 2.1 Presentation of the algorithm

The Quantum Approximate Optimization Algorithm (QAOA, [3]) aims at solving a combinatorial optimization. Given a cost function $C : x \in \{0,\ 1\}^n \to C(x)$, we wish to minimize $C(x)$, ie to find $x^* \in \{0,\ 1\}^n$ such that $x^* = \arg\min_{x\in\{0,\ 1\}^n} C(x)$.

The first idea behind the QAOA is to associate with the cost function $C$ a diagonal hamiltonian $\hat{H}_C = \sum_{x\in\{0,\ 1\}^n} C(x) \left|x\right\rangle \left\langle x\right|$, the eigenstates of which correspond to a particular bitstring $x \in \{0,\ 1\}^n$ and to the associated cost value $C(x)$ as an eigenvalue.
The second idea is taken from adiabatic quantum computing, in which we start with a hamiltonian $H_0$ and a state $\Psi$ which is the ground state of $H_0$, and we slowly transition from $H_0$ to a target hamiltonian $H_1$, with the hope that the state will remain throughout the transition in the ground state of the hamiltonian.
The QAOA operates in a similar fashion, with $H_0 = \sum_1^n X_i$ where $X_i$ is the Pauli operator $X$ acting on qubit $i$, and $H_1 = H_c$, but discretizes the transition between the two hamiltonians. More formally, we prepare the state $\left|+\right\rangle = \left(\frac{\left|0\right\rangle+\left|1\right\rangle}{\sqrt{2}}\right)\otimes^n$ which is the ground state of $H_0 = \sum_1^n X_i$ and we then prepare the state

$$\left|\Phi(\vec{\gamma},\vec{\beta})\right\rangle = \prod_1^p e^{-i\beta_i\hat{H}_0} e^{-i\gamma_i\hat{H}_C} \left|+\right\rangle .$$

Where $\vec{\gamma}$ and $\vec{\beta}$ are parameter vectors which allow us to vary the time of application of each hamiltonian.
Following the ideas of adiabatic computing, for convergence we would expect the first components of beta to be large while those of gamma would be small, and the inverse behavior for the last components. Since we want to chose the best discretization possible, we will prepare this state, then measure its energy, and use a classical optization method to improve the parameters. Convergence is considered reached once the parameters are considered converged. The parameter $p$, the number of discretization steps, is chosen by the user and influences the depth of the circuit as well as the number of parameters to optimize.
Finally, to assess which $x$ is optimal (if we only want to minimal value we may skip this step), we measure several times the value of the observable $\bigotimes Z_i$, and pick the base state which is the most common outcome.

The convergence of the algorithm for a sufficiently high number of layers $p$ is proven in [1] using quantum adiabatic theorems.

### 2.1.1 • Limitations

The problem with the QAOA lies in the fact that each layer requires a considerable number of gates, and even without noise, convergence requires a significant number of layers to ensure a sufficient number of parameters for exploring the entire Hilbert space. In the presence of noise, the depth of the circuits becomes a real issue and convergence is even worse.

The presence of constraints further exacerbates the situation. Penalizing unsuitable solutions within the cost function highlights the lack of degrees of freedom in the optimization process and introduces a tradeoff. On the one hand, penalizing hard enough unwanted solutions allows us to only obtain suitable solutions, but increases the need in variables in order to efficiently separate suboptimal solutions from optimal ones. On the other hand, not penalizing unwanted solutions hard enough results in an optimization process which is slowed by their presence.

One last problem is the apparition of so-called barren plateaus, which are points in the optimization manifold where the gradient vanishes. Those appear when the number of optimization layers increases, and completely hinder the ability of the algorithm to efficiently minimize the energy. Although it is possible with knowledge of a problem to tinker with the ansatz to avoid such plateaus, a very recent article ([2]) discusses the implications of such possibilities, and the fact that they might entail the classical simulability of the algorithm (provided some input is gathered beforehand using a quantum computer), meaning the QAOA would be essentially useless. Finally, there is no proof the QAOA could be faster than classical methods.

# 3
# A Monte-Carlo approach

Our first idea to adress the depth limitations of the QAOA would be to compare its convergence with a Monte-Carlo algorithm which has significantly fewer gates, by replacing $|\Phi(\vec{\gamma},\vec{\beta})\rangle$ with $|\Phi(\vec{\theta})\rangle = \prod_1^n e^{-i\theta_k X_k}|0\rangle$ with a well-chosen $\vec{\theta}_0$. This state requires fewer gates to construct than a single layer of QAOA, and it allows movement throughout the subset of the Hilbert space of interest (as the solution is a state in the basis, it may not be necessary to go through entangled states or even ones with dephasing between $|1\rangle$ and $|0\rangle$). However, it is difficult to estimate which initial parameters would lead to convergence to the optimal solution, because the existence of local minima is very likely (and it is what we have found in our later tests). In light of this our idea was to construct a randomized heuristic, where the initial parameter $\vec{\theta}_0$ is randomly chosen, and we repeat the algorithm, counting each time a particular basis state is obtained. We then choose the one with the lowest energy. Our hope as that when choosing a random starting point on the optimization manifold, the chances of converging towards the ground state energy would be higher than converging towards a local extremum, and thus

repeating the algorithm several times would give us a good chance of finding the ground state. Even though repeating this process a lot might be more costly than using the QAOA, only a single run should be enough to obtain an at least suboptimal solution.
Adding variables means the optimization space is larger, however since the number of variables stays polynomial in the number of qubits, each optimization step will remain polynomial in the number of qubit and thus not a problem on state-of-the-art classical computers.

# 4
# Mixing the two approaches

The preceding approach has the upside of needing very shallow circuits. However since these require no entanglement these are somewhat simpler to simulate on a classical computer ([4]), and since the results are not as convincing as we hoped for (it is a possibility that we never converge towards the ground state, and suboptimal states might be far worse), we wish to improve it. The only way to really improve the algorithm would be to have a mean of determining a good starting point, but doing so would mean basically classicaly solving (by simulating this on a classical computer) the optimization problem itself and isthus out of the scope of our study.
Our idea was therefore to combine the two approaches, by essentially using the QAOA approach but adding degrees of freedom in the mixer hamiltonian to allow our algorithm to explore more efficiently the Hilbert space. More formally, we start from the QAOA but replace $\vec{\beta}$ which is of dimension $p$ (remember that $p$ is the number of layers) with an array $(\beta_{i,j})$ of size $p \cdot n$, and instead of preparing

$$\prod_1^p e^{-i\beta_i \hat{H}_0} e^{-i\gamma_i \hat{H}_C} \left|+\right\rangle ,$$

we prepare the state :

$$\left|\Phi(\vec{\gamma}, \beta)\right\rangle = \prod_1^p e^{-i\sum_0^n \beta_{i,j} X_j} e^{-i\gamma_i \hat{H}_C} \left|+\right\rangle .$$

We introduced more freedom in the optimisation, meaning the energy yielded in the end by this algorithm is going to be better than the one yielded by the QAOA.

# 5
# Traveling Salesman Problem

## 5.1 Presentation of the problem

This subsection is drawn from [5].
The Traveling Salesman Problem (TSP) is an NP-hard combinatorial optimization problem.

Given a list of cities and the distances between each pair of cities, the task is to find the shortest possible route to visit each city exactly once and return to the original city.

Let $n$ be the number of cities, and let $x_{i,t}$ be the variable that is equal to 1 if we are in city $i$ at time $t$ and 0 otherwise. We are interested in the integer TSP problem because our variables take integer values, making our problem a combinatorial optimization problem.

Let $C_{ij}$ be the distances between city $i$ and city $j$.

The objective is to minimize the total distance traveled, meaning we are seeking to find the

$$x^* = \underset{x\in\{0,\ 1\}^n}{\arg\min} \sum_{0\leqslant i,j\leqslant n} C_{i,j} \sum_{t=0}^{n} x_{i,t} x_{j,t+1}$$

with the condition that we can only visit a given city once:

$$\sum_{0\leqslant i\leqslant n} x_{i,t} = 1$$

and we can only be at one city at a given time:

$$\sum_{0\leqslant t\leqslant n} x_{i,t} = 1$$

.

## 5.2 Formulation compatible with the QAOA

In order to use the QAOA, we need to define the cost function

$$C = C_{\text{cost}} + M \cdot C_{\text{penalty}}.$$

The cost function $C$ is composed of the function to minimise

$$C_{\text{cost}}(x) = \sum_{0\leqslant i,j\leqslant n} C_{ij} \sum_{t=0}^{n} x_{it} x_{j,t+1}$$

and the penality function in order to respect the constraints imposed:

$$C_{\text{penalty}} = \sum_{t} \left(1 - \sum_{i} x_{i,t}\right)^2 + \sum_{i} \left(1 - \sum_{t} x_{i,t}\right)^2$$

(note that $C_{\text{penalty}}(x) = 0$ iff $x$ respects the constraints). $M$ is a constant large enough for the penalization to be efficient.

We note the cost Hamiltonian :

$$\hat{H}_C = \sum_{x\in\{0,\ 1\}^n} C(x) \left|x\right\rangle \left\langle x\right|$$

The quantum state is:

$$\left|\Phi(\vec{\gamma}, \vec{\beta})\right\rangle = \prod_{1}^{p} e^{-i\beta_i \hat{H}_B} e^{-i\gamma_i \hat{H}_C} \left|+\right\rangle.$$

Where $\hat{H}_B = \sum_1^n X_i$ and $\left|+\right\rangle = \left(\frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) \otimes^n$

## 5.3 Results

We placed ourserlves in the case where we have 3 cities, $M = 6000$ (this value will be discussed) and the cost list is $C = [[0, 1000, 1], [1, 0, 1000], [1000, 1, 0]]$, which means that the best way when starting from city 1 to return to city 1 is to go from city 1 to city 3, then to city 2, and then return to city 1 (if the starting city changes, any circular permutation of this order is optimal). The state is represented by a ket that describes $x_{ij}$, for example if we have the ket $|100010001\rangle$ that means that at time t=1, we are at city 1, at time t=2 we are at city 2, and at time t=3, we are at city 3.

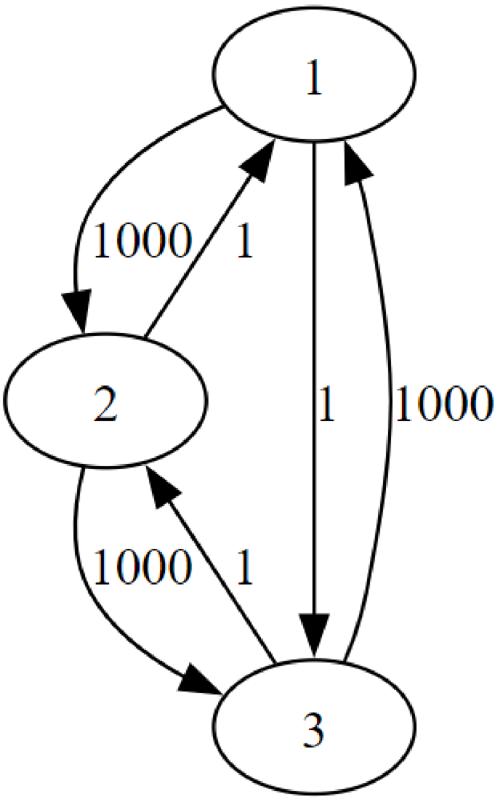

Figure 1: Map of the cities with displacement cost

### 5.3.1 • Results using the QAOA

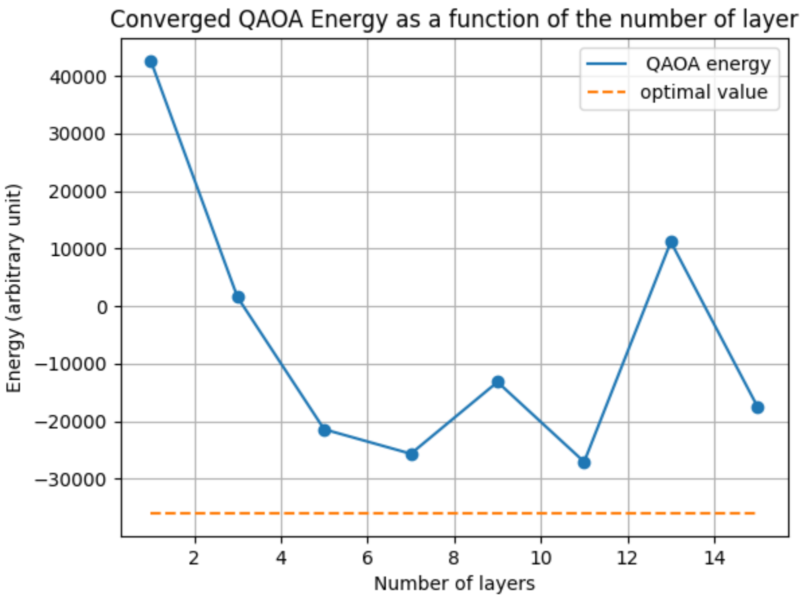

Figure 2: Converged QAOA energy as a function of the number of layers $p$. Parameters for the optimization were a convergence threshold of $10^{-5}$ and a maximum number of iterations of 10000. Initial parameters were chosen all equal to 0.1.

What we can see is that even with a large number of layers, the algorithm is far from being close to the optimal energy. This means that if we wanted to retrieve the optimal state (which corresponds to the optimal path) we would not get a decisive answer (and any answer would likely be mediocre). Starting from $p = 9$, we see the appearance of the barren-plateau phenomena, which leads to worse results. Had we increased significantly the number of layers (which sadly goes beyond the capabilities of our computers), without encounterring barren-plateaus, we would have probably seen the difficulty of efficiently separating optimal solutions from sub-optimal ones.
As the public (free) version of Eviden's framework Qaptiva does not include noise models, we have not been able to test the algorithm in the presence of noise. It is however likely that, even with noise levels below what is currently attainable, the results would have been highly mediocre.

### 5.3.2 • Results with Monte-Carlo approach

We ran our algorithm 200 times to have a better idea of the distribution of states obtainable. In all our tests, our wavefunction converged towards a basis state, confirming our initial intuition.

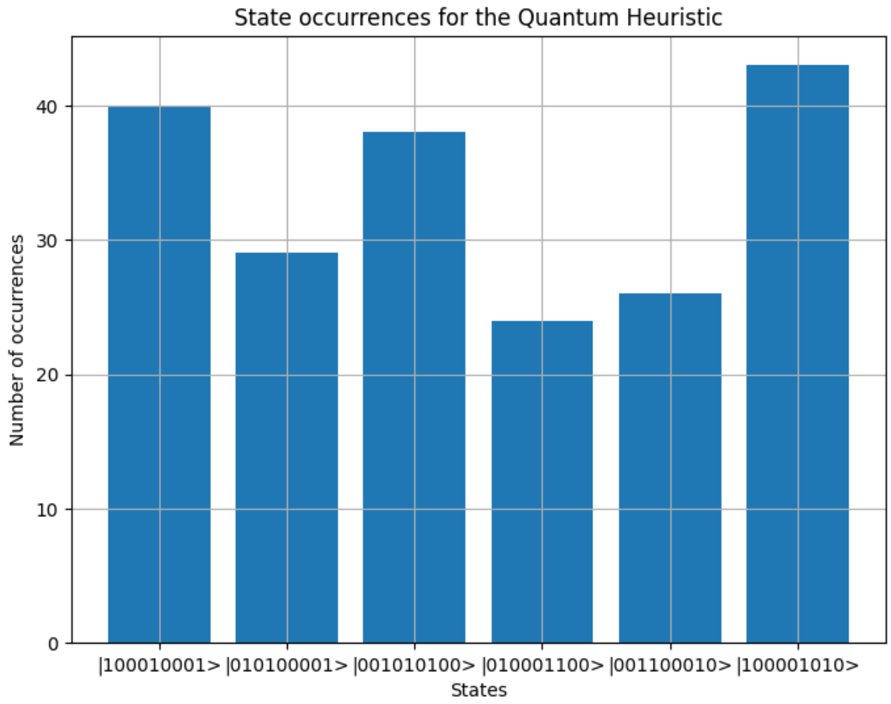

Figure 3: Occurrences of states using our monte-carlo algorithm. The number of shots used was 10000, and the maximum number of iterations was 150.

We found that the state of maximum occurrence (in this case $|100001010\rangle$, meaning we start at city 1, then go to city 3, and then to city 2) is indeed an optimal state. However, several suboptimal states are highly represented as we feared, meaning it is possible to never converge towards an optimal state

### 5.3.3 • Energy Comparison

Here we compare the energy yielded by the QAOA with an increasing number of layers with the energy yielded by a single run of our monte-carlo each time.

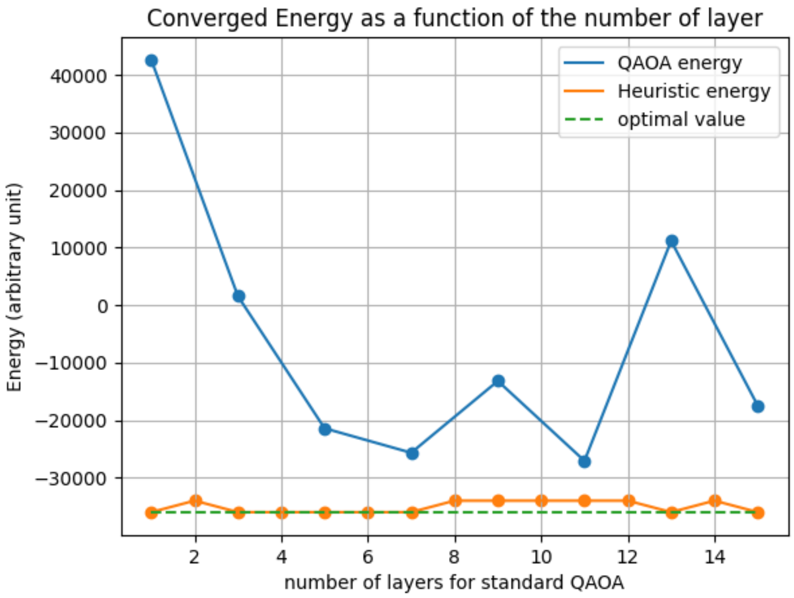

Figure 4: Comparison between the energy values yielded by the QAOA and the energies yielded by single runs of our Monte-Carlo approach

This comparison shows that even though our Monte-Carlo algorithm does not only yield optimal states, The energies of the states it yields are significantly better than those produced by the QAOA. This means that our random heuristic (during which we run the heuristic several times, and then choose the state with the lowest energy) is likely to perform better than the QAOA. There however are limitations that are common to all Monte Carlo methods, which are that in order to ensure convergence we need an infinite amount of runs. In a case where the suboptimal solutions are highly worse than the optimal ones, a Monte Carlo without a high enough number of runs would yield mediocre results.

### 5.3.4 • Results of our third approach

Since the increase in variables is substantial, we have only been able to test our approach with 3 layers. It however yielded a base state which is either optimal or suboptimal (which is of better energy than the QAOA result). The mathematical convergence of QAOA also applies to our algorithm (since the par of the hilbert space accessible by a QAOA ansatz is included in that accessible by our algorithm). This approach therefore combines both a better accessible space as well as a proven convergence for a large enough amount of layers.

# 6 CONCLUSION

Our simulations highlight the limitations of the Quantum Approximate Algorithm, as well as simple (even if costly) ways to improve it. Performing noisy simulations of our algorithms could allow us to also garner a better understanding of how to enhance the algorithm's noise resilience as well as understanding better the relationship between noise resilience and size of the classical optimization landscape, which is the problem lying at the core of the NISQ approach.

# REFERENCES

[1] Lennart Binkowski, Gereon Koßmann, Timo Ziegler, and René Schwonnek. Elementary proof of qaoa convergence, 2023.

[2] M. Cerezo, Martin Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana, Manuel S. Rudolph, Pablo Bermejo, Aroosa Ijaz, Supanut Thanasilp, Eric R. Anschuetz, and Zoë Holmes. Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing, 2023.

[3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[4] Maxime Oliva. An entanglement-aware quantum computer simulation algorithm, 2023.

[5] Wenyang Qian, Robert A. M. Basili, Mary Mehrnoosh Eshaghian-Wilner, Ashfaq Khokhar, Glenn Luecke, and James P. Vary. Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem. *Entropy*, 25(8):1238, August 2023.