Qualifying quantum approaches for hard industrial optimization problems. A case study in the field of smart-charging of electric vehicles

Constantin Dalyac[1,2], Loïc Henriet[*1], Emmanuel Jeandel[3], Wolfgang Lechner[4,5], Simon Perdrix[3], Marc Porcheron[6], and Margarita Veshchezerova[3,6]

[1]Pasqal, 2 avenue Augustin Fresnel, 91120 Palaiseau, France
[2]LIP6, CNRS, Sorbonne Université, 4 Place Jussieu, 75005 Paris, France
[3]LORIA, CNRS, Université de Lorraine, Inria Mocqua, F 54000 Nancy, France
[4]University of Innsbruck, Technikerstrasse 21a, Innsbruck, Austria
[5]Parity Quantum Computing GmbH, Rennweg 1, Innsbruck, Austria
[6]EDF R&D, 7, boulevard Gaspard Monge, 91120 Palaiseau, France

January 1, 2021

Abstract

In order to qualify quantum algorithms for industrial NP-Hard problems, comparing them to available polynomial approximate classical algorithms and not only to exact ones – exponential by nature – , is necessary. This is a great challenge as, in many cases, bounds on the reachable approximation ratios exist according to some highly-trusted conjectures of Complexity Theory. An interesting setup for such qualification is thus to focus on particular instances of these problems known to be "less difficult" than the worst-case ones and for which the above bounds can be out-performed: quantum algorithms should perform at least as well as the conventional approximate ones on these instances, up to very large sizes. We present a case study of such a protocol for two industrial problems drawn from the strongly developing field of smart-charging of electric vehicles. Tailored implementations of the Quantum Approximate Optimization Algorithm (QAOA) have been developed for both problems, and tested numerically with classical resources either by emulation of Pasqal's Rydberg atom based quantum device or using Atos Quantum Learning Machine. In both cases, quantum algorithms exhibit the same approximation ratios than conventional approximation algorithms, or improve them. These are very encouraging results, although still for instances of limited size as allowed by studies on classical computing resources. The next step will be to confirm them on larger instances, on actual devices, and for more complex versions of the problems addressed.

1 Introduction

In the quest for a quantum advantage, situating quantum algorithms and technologies with respect to classical approaches for solving NP-hard problems is a key question. Seminal works from Shor and Grover were a major breakthrough in this perspective in the early developments of quantum computing. Based on the theoretical model of quantum computation, they developed quantum algorithms that provide respectively an exponential speed-up for integer factorization [1], and a quadratic speed-up on exhaustive search in an unstructured database [2], a principle applicable to any NP problem providing

---

[*]corresponding author : loic@pasqal.io

at most such a speed-up in terms of query complexity [3, 4]. However, implementing these algorithms requires fault-tolerant quantum machines handling a large number of qubits, which have yet to be built.

While many technological obstacles currently impede the creation of such machines, experimental physicists have been capable of controlling quantum systems precisely enough to simulate complex many-body quantum systems. These quantum devices present strong quantum properties and offer scientists a control on the quantum aspects of physical systems. They can have sizes of several hundreds of quantum particles, and because of the unavoidable coupling between the system and its environment, these quantum platforms fall in the category of Noisy Intermediate Scale Quantum (NISQ) devices [5]. Among them, there is a strong belief that Analog Quantum Simulators (AQS) can perform specific tasks intractable for classical computers in polynomial time, such as the dynamical simulation of strongly interacting quantum Hamiltonians [6, 7], and it is expected that AQS will be among the first to propose useful applications in the short-term [8]. Lately, there has been growing interest in knowing if the quantum characteristics of these devices can be steered towards outperforming classical computers on industry-relevant tasks. An active field of research is currently guided towards combinatorial optimization, where the Hilbert space spanned by the many-body quantum system is used to efficiently encode a high-dimensional discrete problem.

In this context, algorithms that can run on such NISQ devices have been developed. Quantum Annealing [9], the Quantum Adiabatic Algorithm [10], and the Quantum Approximate Optimization Algorithm [11] are among the most promising ones. A better understanding of the performances of these approaches on industrial NP-hard problems is of great interest, both for quantum computing adoption and for the application domains concerned.

Approximate results are of great interest for practical applications, specifically industrial ones where a close-to-optimal solution is of significant value when exact solutions are unreachable due to exponential conventional computing time. Interestingly, some NP-hard problems are easier to approximate than others. For example, while the 0/1-Knapsack problem is NP-complete, it admits a fully polynomial-time approximation scheme (PTAS) [12]. To grasp the performances and the quality of solutions provided by NISQ algorithms, we must rigorously compare them to their approximate classical counterparts. Determining where quantum approaches sit in the approximation complexity landscape is key in understanding their potential for practical problems. One should be aware that for some NP-hard problems bounds have been proven on these ratios which cannot be exceeded by classical algorithms unless some very strongly-believed conjectures of Complexity Theory would be broken; thus, improving these ratios beyond these bounds in these cases would establish a quantum supremacy for solving some NP-hard problems. Another desirable scenario is that quantum approaches find comparable approximation ratios as classical algorithms but faster or at a lower cost, which would already provide significant value for industries.

Taking this latter point into consideration, an interesting workout for qualification of quantum approaches is to focus on particular instances of NP-Hard problems known to be "less-difficult" than the worst-case ones, in the sense that approximation algorithms do exist for them that could outperform the above bounds, which are precisely established for the worst-case instances : our quantum algorithms should perform at least as well as the conventional approximate ones on these instances, up to very large sizes.

In this article, we present and discuss a case study based on this protocol, for two problems drawn from the rapidly growing sector of smart-charging of electrical vehicles in which EDF, the French utility for electricity production and supply, is strongly involved. Once appropriately modeled, these

problems appear as classical NP-hard graph theory problems, Max-k-Cut and Maximum Independent Set (MIS) respectively. To solve these problems, we develop dedicated extensions of the Quantum Approximate Optimization Algorithm (QAOA) [11, 13]. Besides providing numerical results evaluating the performances of these approaches on real data sets of electrical vehicle loads, the aim of this paper is to illustrate practical issues faced when trying to address real industrial problems with available NISQ frameworks. In particular, achieving good performances with quantum approaches requires the design of hardware-efficient procedures, that exploit the strengths of a given quantum processor. In this framework, hardware and software are jointly developed in order to optimize the execution of the overall implementation. Among the variety of platforms that are currently being investigated, programmable arrays of single neutral atoms manipulated by light beams appear as a very powerful and scalable technology to manipulate up to a few thousands qubits [14–16]. For the problems under consideration, we provide some implementation details on such platform as developed and commercialized by the company Pasqal.

The paper is organized as follows : section 2 describes the smart-charging of electrical vehicles problems (SC1) and (SC2) which serve as use-cases ; section 3.2 and 3.3 present the extension of QAOA to Max-k-Cut and the implementation of the approach proposed in Ref. [13] to solve MIS, respectively ; section 4 is dedicated to our results and section 5 to conclusions and further works. Complementary material is presented in the appendix.

2 Two smart-charging problems and their modeling as NP-hard problems

It is admitted that in the next decades Electric Mobility will play a key role to solve major environmental and public health problems, thanks to the reduction of greenhouse gas and fine particles emissions it allows. Smart-charging appears as a mandatory condition to allow electric mobility expansion. Besides, the use of vehicle batteries both as energy storage and power supply devices ("Vehicle to Grid" or "V2G") could significantly improve the flexibility of the electric system, reducing high-peak of electricity demand thus saving significant energy.

Many difficult problems lie behind this scheme in order to optimize the management of the electric system in terms of cost while satisfying various hard technical constraints. These include, among others, the modulation of electricity demand taking into account the potentially high demand specific to electric vehicle loads, the needs of electric vehicle users, the charging/discharging cycles of vehicle batteries and the reserves required to guarantee the frequency stability of the grid.

A great deal of theses problems takes the form of typical *scheduling* and *Operational Research* problems. They are large sized combinatorial optimization problems, many of them known to be NP-hard/complete.

The following sections describe two samples of these problems, and their modeling for quantum resolution.

2.1 Vocabulary and common hypotheses

Both problems will be tackled under the following assumptions:

- A load station is made up of several charging points, each of them loading at most a single electric vehicle (EV) at a given time step;

3

- The charging points are *parallel identical* machines that supply the same power. The charging time of a given EV is thus independent of the charging point it is scheduled on;

- We consider neither additional job characteristics and constraints (release/due dates, charging profile imposed by the battery state) nor global resource constraints on the load station (maximal power deliverable at a given time step);

- Preemption is not allowed: a load task cannot be interrupted to be resumed later, on the same charging point or another one.

2.2 Minimization of Total Weighted Load Completion Time (SC1) and Max-k-Cut

We consider $J = \{1, \ldots, n\}$ charging jobs of n EVs with durations $T = \{t_1, \ldots, t_n\}$, to be scheduled on a set $I = \{1, \ldots, k\}$ of k charging points. An integer weight $w_j > 0$ is associated to each job j, measuring its importance. For example, we want to prioritize the charge of safety-related intervention vehicles. The time at which a load j ends, called the *completion time* is noted C_j and we want to minimize the weighted total time of completion of the charges $\sum_{j \in J} w_j C_j$.

(SC1) is a classical scheduling problem known to be NP-hard in the general case, polynomial on a single machine or without priorities/weights attached to the jobs [17]. If the number of machines m is fixed, (SC1) is NP-hard in the weak sense that it can be solved by pseudo-polynomial algorithms[1], typically based on dynamic programming. In the case where m is not fixed, (SC1) is NP-Hard in the strong sense, meaning that no such pseudo-polynomial algorithm exists except if P=NP.

Different approximation approaches have been studied for (SC1), until some PTAS in the general case and FPTAS[2] in the weakly NP-Hard variant were established [18–21].
Interestingly for our purpose, some of these approximation approaches were based on the reformulation of (SC1) as a weighted Max-k-Cut problem, a problem that can be tackled by the Quantum Approximation Optimization Algorithm (QAOA), at least in its $k = 2$ set up i.e. Max-Cut.

Consider the complete graph $G = (V, E)$ whose vertices V correspond to the n jobs in J, and with a weight assigned to each edge (k, j) in E defined by $w_{kj} = \min\{w_k t_j; w_j t_k\}$. The maximal k-cut of this graph provides with the optimal affectation of the n jobs on the k machines. Informally, this relies on the well known "Smith Rule" [22] which states that once jobs have been affected to machines, the optimal scheduling is given by executing them in a non-increasing order defined by the ratio $\frac{w_j}{t_j}$. On this basis, minimizing the weighted total completion time of the tasks can be shown to be equivalent to maximizing the above weight on crossing edges between the subsets of a m-partition of V [18, 19].

For both Max-Cut and Max-k-Cut, known polynomial randomized approximation algorithms obtain high *approximation ratios* C/C_{opt} – where C represents the average value of the solution provided by the algorithm and C_{opt} the optimal one –, namely 0.878567 [23] and $\left(1 - \frac{1}{k} + 2\frac{\ln k}{k^2}\right)$ [24], respectively (the latter is improved for small values of m in Ref. [25]). Moreover, improving these ratios is proved to be NP-Hard, unless some highly-believed conjectures of Complexity Theory would be false. In the case of Max-Cut, improving the approximation ratio from 0.878567 up to ($\frac{16}{17} = 0.941176$) is NP-Hard [26] and thus out of reach of any polynomial classical algorithms unless $P = NP$, as it is to increase the ratio upon $1 - 1/34k$ for Max-k-Cut [27]. Likewise, both problems are APX-Hard and thus, unless $P = NP$, they have no PTAS [24, 28]. Finally, tighter results were proven under the "Unique Games

---

[1]An algorithm is said to be *pseudo-polynomial* if it is polynomial in the numeric values of its data, but super-polynomial in the length of their binary encoding.

[2]A *Polynomial Time Approximation Scheme* (PTAS) is a family of ε-parametrized polynomial algorithms allowing to approximate optimal solutions "arbitrary closely" by a factor of $(1 - \varepsilon)$, $\varepsilon > 0$. When the algorithms are also polynomial in their parameter ε, one talks of *Full Polynomial Time Approximation Scheme* (FPTAS)

Conjecture", namely that improving the above original ratios for Max-Cut and Max-k-Cut established in Ref. [23] and Ref. [24] is NP-Hard [29].

This means that such improvements by some quantum algorithms would establish a "quantum supremacy" on NP-Complete problems, unless these conjectures turn out to be false, which is considered unlikely. Here, we do not aim at achieving such an approximation ratio improvement with quantum algorithms in the general/worst-case. Alternately, we analyze their performances on some graphs drawn from real-world problems considering the performances obtained by the best-known randomized approximation classical algorithms. We observe that QAOA outperforms Goemans and Williamson ratio on these particular instances of Max-Cut. Similar results were observed in Ref. [30] on small random graphs. Establishing that such improvement would hold true for worst-case and large graphs instances is still an open question, and will remain so as long as we stay in the NISQ era. In any case, reaching similar approximation ratios as classical solutions, but faster or at a lower energy cost would already be of significant value for industries.

In this context, it should be noted that although these limits to improving the approximation ratios are valid for general/worst-case instances of Max-k-Cut, it does not mean that they are valid for particular instances with a specific structure. Thus, since efficient classical approximation schemes are known for (SC1), we can expect its instances reformulated as instances of Max-k-Cut to be easier to approximate than general/worst-case ones. We will see that this is indeed the case, both when using classical and quantum algorithms.

2.3 Optimal Scheduling of Load Time Intervals within Groups (SC2) and MIS

We now consider the following problem (SC2): given a set of load tasks represented as *intervals* on a timeline, such that each of them belongs to a specific *group* , for example distinct vehicle fleets of a company, select a subset of these loads (i) which maximizes the number of non-overlapping tasks and (ii) such that at most one load in each group is completed. The goal is here to both minimize the completion time of the selected loads and to guarantee that no group will be over-represented in the schedule.

This problem belongs to the class of *Interval Scheduling* problems [31]. More precisely, it is a *Group Interval Scheduling*, or *Job Interval Selection* problem. It can be restricted without loss of generality to the case where all the groups contain the same number of tasks, k. It is NP-Complete for $k \geq 3$, and has no PTAS for $k \geq 2$ unless $P = NP$ [32]. Some polynomial approximation ratios have been obtained in the general case, namely 0.5 in Ref. [32], improved to 0.63211 in Ref. [33], while polynomial algorithms exist in cases where some parameters are fixed [34].

Let $I = \{(s_1, e_1) \ldots (s_{nk}, e_{nk})\}$, where n is the number of groups, be the set of intervals representing load job starting and ending dates, and $G = (V, E)$ be the graph whose vertices in V correspond to intervals in I, and with an edge (i, j) in E *iff* interval i and j overlap **or** i and j belong to the same group. Clearly, an *independent set* of this graph, i.e. a set of vertices no two of which are adjacent, represent a feasible solution of the problem, and its Maximum Independent Set (MIS) is the optimal one[3].

Following our protocol for quantum algorithm qualification, we will limit ourselves to specific instances of (SC2) that can be formulated as MIS on two-dimensional *Unit-Disk* (UD) graphs. These geometrical graph are graphs in which two vertices are coupled by an edge if the distance between them

---

[3]The above formulation supposes that a *starting date* is affected to each load task, in order to represent it as an interval on the time line. In a real smart-charging management system, such dates could be fixed by the users of the vehicles, imposed by technical constraints or decided by the smart-charging manager.

is below a threshold value. This choice is motivated by two reasons. First, as described at section 3.3 below, Pasqal's neutral atom quantum processor is particularly well suited to natively implement the MIS on Unit-Disk graphs. Second, the MIS on such graphs, although remaining NP-Complete, is known to be "less difficult" to approximate than the MIS on general graphs, and has a PTAS while the general MIS does not [35, 36]. Of course, this protocol requires to transform (SC2) graphs to Unit-Disk graphs, a procedure we discussed in section 3.3.2.

3   Quantum approaches to smart-charging problems

In this section, we describe how to use quantum approaches for solving the problems presented above.

3.1   QAOA in a nutshell

The "Quantum Approximate Optimization Algorithm" (QAOA) computes approximate solutions to combinatorial optimization problems, with a theoretical guarantee of convergence when the depth of the quantum circuit increases [11].

QAOA is a variational algorithm for combinatorial problems in which a quantum processor works hand-in-hand with a classical counterpart, as illustrated in Fig. 1 (see Ref. [37] for a review on variational algorithm). The quantum processor is used to prepare a wave function $|z_{\gamma,\beta}\rangle$. In the most general case, $|z\rangle = |z_1 z_2 ... z_n\rangle$ represents a n-qudit state vector, with $z_i \in \{0, 1, .., d\}$, and the subscript in $|z_{\gamma,\beta}\rangle$ indicates that the state belongs to a family of states that is parameterized by the angles γ and β. More specifically, $|z_{\gamma,\beta}\rangle$ is generated by the successive application of unitaries generated by the non-commutative operators $\hat{M}$ and $\hat{C}$, with angles given by $\beta = (\beta_1, \beta_2, ..., \beta_p)$ and $\gamma = (\gamma_1, \gamma_2, ..., \gamma_p)$, respectively. Given an initial state $|z_0\rangle$, the wavefunction prepared by the quantum processor takes the following form,

$$|z_{\gamma,\beta}\rangle = e^{-i\beta_p \hat{M}} e^{-i\gamma_p \hat{C}} \ldots e^{-i\beta_1 \hat{M}} e^{-i\gamma_1 \hat{C}} |z_0\rangle. \tag{1}$$

A common choice for the cost operator $\hat{C}$ (also sometimes referred to as energy operator) is the diagonal operator in the computational basis, $\hat{C}|z\rangle = C(z)|z\rangle$, where $C(.)$ is the cost function to be optimized for, while the mixing operator $\hat{M}$ induces transitions between states in the computational basis [11, 38]. In the following, we will always reformulate our problems under the form of minimization problems, by changing the sign of the cost function of the original problems. The ground state of the energy operator $\hat{C}$ corresponds to the optimal solution to the optimization problem. The dimension p of the vectors γ and β is called the depth of the algorithm.

The quantum state is then measured to construct an statistical estimator $\langle \hat{C} \rangle_{\text{obs}}$ of the cost function to be minimized. A classical optimization procedure uses this estimator to update the variational parameters γ and β for the next iteration. This loop repeats until convergence to a final state, from which an estimate of the solution to the problem is extracted. The approximation ratio $\langle \hat{C} \rangle / C_{\text{opt}}$, where $C_{\text{opt}} = \min_z C(\mathbf{z})$, measures the quality of the approximation yielded by QAOA.

In the following, we will consider two distinct ways to prepare the ansatz wavefunction (1) on Pasqal quantum processors. In the first one, the quantum processor is used in digital mode, and the trial wavefunction is the output of a quantum circuit composed of discrete quantum gates. In the second one, the quantum processor is used in an analog manner and the trial wavefunction results from the application of a continuously parameterized Hamiltonian.
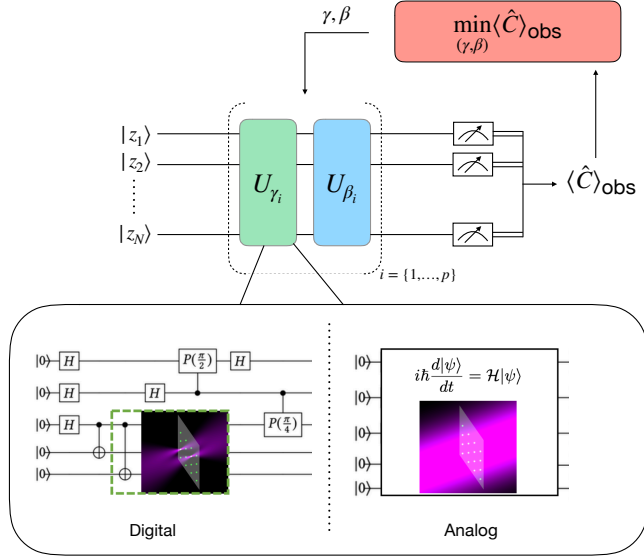
Figure 1: Principle of the QAOA algorithm. A quantum processor, which can be operated in either digital or analog mode, is used to prepare an ansatz wavefunction from which we construct the mean value $\langle \hat{C} \rangle_{\text{obs}}$ using numerous measurements. A classical optimizer then updates the variational parameters. Some problems are naturally tailored to the analog mode of the platform, while others require a digital mapping. In digital mode, the unitaries are built from quantum circuits, made of elementary quantum gates acting each on one or a few qubits. In the analog mode, the unitaries are built from sequences of Hamiltonians that can be controlled in a continuous manner.

3.2 Max-k-Cut

We describe here an extension of QAOA to deal with weighted Max-k-Cut problems and using quantum circuits. We also introduce a hardware-efficient implementation of this procedure on arrays of neutral atoms.

3.2.1 Encoding on the quantum processor

We show here how to extend QAOA to Max-k-Cut problems on weighted graphs. Denoting by $z_i \in \{1, 2, .., k\}$ the subset node i belongs to, the goal of Max-k-Cut is to minimize the cost function:

$$C(z_1, \ldots, z_N) = -\sum_{u=1}^{N} \sum_{v=u}^{N} w_{uv}(1 - \delta_{z_u, z_v}), \tag{2}$$

where the edge w_{uv} participates to the cost function if nodes u and v are in separated subsets.

Originally QAOA was designed for optimization problems of type QUBO (Quadratic unconstrained binary optimization). In that case, a problem with n variables naturally maps to a system of n qubits. For optimization over integer-valued variables one has to define a correspondence between basis states of a Hilbert space of some dimension and solutions to the initial problem, and then restrict the quantum evolution to the feasible domain.

A possible approach to tackle this issue is to apply the unary encoding [38] that for any natural $k > 2$ attributes k qubits to each node and a node u is colored in c if $|z_u\rangle = |0\ldots010\ldots0\rangle$ where the bitstring z_u has only one 1 on the position c. This encoding uses Nk qubits for an instance with

7

N nodes and requires some non-trivial modifications to the mixing operator M in order to keep the evolution in the feasible subspace, which is spanned by basis states with only one 1 per node. This encoding was also used in Ref. [39] for solving graph-coloring problem with QAA.

We suggest instead to use the conventional binary encoding of integers for $k = 2^l$. In this encoding each node is associated to a set of l qubits $\mathbf{z} = \{z^{(0)}, \ldots, z^{(l-1)}\}$ that will indicate the color of the node. More specifically, having qubits in the state $\mathbf{z}_u$ corresponds to having the node u in the subset $z_u = z^{(0)}z^{(1)}...z^{(l-1)}$. Note that we need $Nl \approx N \ln k$ qubits to encode the entire coloring of the graph. In the specific case where $k = 2^l$, the computational basis that spans the Hilbert space of our platform corresponds exactly to all the 2^{Nl} possible colorings of the graph. It means that, contrary to the unary encoding, no modification to the operator M is required. This approach is easier to compile to elementary gates and less expensive in terms of number of qubits than the unary encoding. A natural extension for all natural k would be to use qudits instead of qubits [40].

For binary encoding we can build a cost operator $\hat{C}$ that is diagonal in the computational basis, such that $\hat{C}|\mathbf{z}_1 \ldots \mathbf{z}_N\rangle = C(\mathbf{z}_1 \ldots \mathbf{z}_N)|\mathbf{z}_1 \ldots \mathbf{z}_N\rangle$, by writing :

$$\hat{C} = -\sum_{1 \leq u < v \leq N} w_{uv} \left(1 - \frac{1 + \sigma_u^{(0)}\sigma_v^{(0)}}{2} \cdots \frac{1 + \sigma_u^{(l-1)}\sigma_v^{(l-1)}}{2}\right), \tag{3}$$

where $\sigma_u^{(i)}$ corresponds to a Pauli-Z matrix σ_z acting on the atom $i \in \{0, \ldots, l-1\}$ associated to node u. The operator C in Eq. (3) can be decomposed as a sum of operators with $\{2, 4, \ldots, 2l\}$-body interaction terms. Interaction terms involving more than 2-body operators are not directly implementable on most quantum computing platforms which only support 2-qubit gates. Instead, they can be decomposed as sums of two-body terms, which can be realized with CNOT gates. An example of the resulting circuit for an edge is shown in Fig. 2.
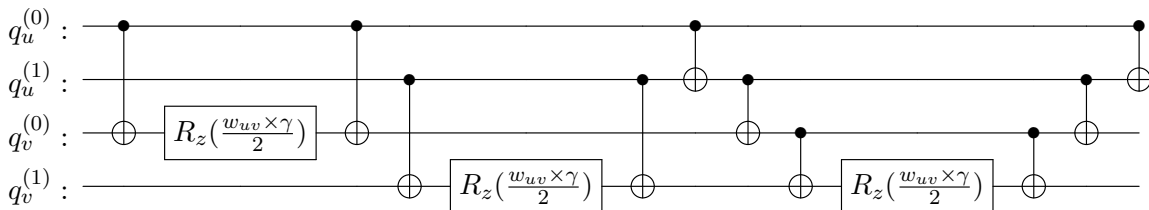

Figure 2: A quantum circuit implementing a term $e^{-i\gamma C_{u,v}}$ corresponding to the edge $\langle u, v \rangle$ in binary encoding for $k = 4$

Because of the completeness of the Max-k-Cut graphs that we are studying, the cost operator in Eq. (3) involves coupling terms between all qubits. However, all quantum chips have a finite connectivity in practice. The physical realization of the desired terms between remote qubits thus requires the introduction of a large number of SWAP gates, that are detrimental to the performance of the procedure. Next, we introduce a hardware-efficient implementation of Max-k-Cut on Rydberg atom arrays which minimizes this overhead.

3.2.2 Towards a hardware-efficient implementation of Max-k-Cut on Rydberg atom arrays

Mapping the QUBO representation of the Max-k-Cut problem as discussed above introduces on the one hand an overhead in number of CNOT gates as a result of the decomposition of multi-spin terms into pair interactions. An alternative to the QUBO representation of the problem is the parity encoding,

originally proposed in Ref. [41] (Lechner-Hauke-Zoller scheme) and recently extended to applications using QAOA [42].

Let us summarize here the main steps of the parity transformation for completeness and then apply it to the Max-k-Cut problem. The parity encoding encodes the relative orientation of several logical qubits as a single physical qubit. With this encoding, each physical qubit represents the parity of several logical qubits, e.g. the physical spin $\hat{z}_{12} = z_1 z_2$ represents the product of two logical bits in the QUBO representation. Here, $\hat{z}_{ij}$ is used as short notation of the z-component of the physical qubit $\sigma_{ij}^{(z)}$ and z_i is used for the the logical qubits $\sigma_i^{(z)}$. The parity encoding represents the full problem in terms of the parity variables. This is not trivially possible, as the number of degrees of freedom of the logical graph are N (the number of logical spins), while the number of physical spins is K, the number of edges in the graph. To compensate for this difference, a number of $C = K - N + 1$ constraints need to be introduced. These constraints are conditions on the physical qubits that have to be fulfilled. In the Lechner-Hauke-Zoller scheme, these constraints are constructed from closed loops in the logical graph which results in a physical implementation that consists of K qubits and $K - N + 1$ 4-body constraints. Remarkably, the problem is fully encoded in the local fields while all interactions are uniform and problem independent [41].
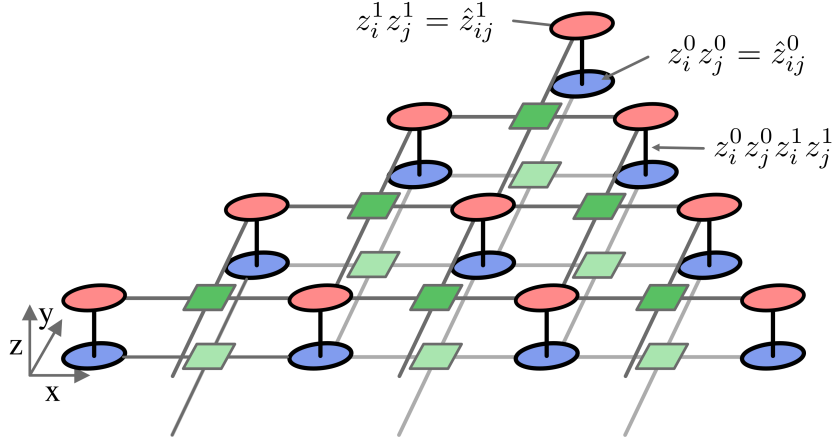

Figure 3: Parity transformation of the Max-4-cut problem. The setup consists of two layers of qubits (blue bottom and red top). The bottom layer represents the 0th bit of the color and the top layer the 1st bit of the color. The qubits within the same layer interact via 4-body constraints [41] (green squares). The interaction terms $\hat{z}_i^0 \hat{z}_i^1 = z_i^0 z_j^0 z_i^1 z_j^1$ are efficiently encoded as pair gates between neighboring qubits of the two layers (black lines). Note, that the 4-body constraints are decomposed into parallelizable CNOT gates within the layers [42].

Applying the parity encoding to the Max-4-cut problem allows us to reduce the number of CNOT gates to order N^2. In the first step, we make use of the parity transformation of each term $z_i^0 z_j^0 \to \hat{z}_{ij}^0$ and $z_i^1 z_j^1 \to \hat{z}_{ij}^1$. As a second step we transform the 4-body term of the two-coloring problem which has the form $z_i^0 z_j^0 z_i^1 z_j^1$. We notice, that this term in the parity encoding is $\hat{z}_i^0 \hat{z}_i^1$ and thus a simple pair interaction between two physical qubit. Remarkably, the pair interaction can be performed between nearest neighbors by introducing a two-layer setup as depicted in Fig. 3.

The cost function using the above construction reads as

$$C = \sum_i \left(\hat{z}_i^1 + \hat{z}_i^0 + \hat{z}_i^0 \hat{z}_i^1 \right) + \sum_l C_4 \hat{z}_i^0 \hat{z}_j^0 \hat{z}_k^0 \hat{z}_l^0 \tag{4}$$

9

Let us now comment on the resources requirements for this hardware-efficient implementation. The number of qubits is $N(N-1)$ and the number of CNOT gates is $9N^2 - 41N + 48$. In the above example for $N = 5$ the number of qubits is $N_p = 10$ and number of CNOT Gates is $N_c = 68$. In addition, the CNOT gates can be fully parallelized which results in a constant depth of global gates. In comparison, the number of qubits in the standard gate model is N and the number of CNOT gates to be calculated but assumed to be order N^3. This encoding provides thus a strong advantage in platforms which show a capacity to scale up the number of qubits.

With a number of thousands of qubits of next generation neutral atom devices [16] in combination with the prospect of full parallelizability of the parity encoding, Max-k-Cut represents a realistic application of near term quantum devices.

3.3 MIS

In this Section we present how to solve a MIS problem using a QAOA procedure on a neutral atom processor working in an analog mode [13, 43]. We first show how an analog control over the atoms allows us to realize a Hamiltonian reproducing the MIS cost function in the case of Unit Disks (UD) graphs. Then, we discuss a procedure enabling us to transform more generic interval graphs to UD graphs.

3.3.1 Rydberg blockade and graph independent sets

When looking for the MIS of a graph, we separate the nodes into two distinct classes: an independence one and the others. We can attribute a status z to each node, where $z_i = 1$ if node i is attributed to the independent set, and $z_i = 0$ otherwise. The Maximum Independent Set corresponds to the minima of the following cost function:

$$C(z_1, \ldots, z_N) = -\sum_{i=1}^{N} z_i + U \sum_{\langle i,j \rangle} z_i z_j \tag{5}$$

where $U \gg 1$ and $\langle i,j \rangle$ represents adjacent nodes (*i.e.* there is a link between node i and j). In this cost function, we want to promote a maximal number of atoms to the 1 state, but the fact that $U \gg 1$ strongly penalizes two adjacent vertices in state 1. The minimum of $C(z_0, \ldots, z_N)$ therefore corresponds to the maximum independent set of the graph.

Interestingly, the operator $\hat{C}$ associated with the cost function of Eq. (5) can be natively realized on a neutral atom platform [13], with some constraints on the graph edges. We map a ground state and a Rydberg state of each atom to a spin 1/2, where $|1\rangle = |r\rangle$ is a Rydberg state and $|0\rangle = |g\rangle$ is a ground state. An atom in a Rydberg state has an excited electron with a very high principal quantum number and therefore exhibits a huge electric dipole moment. As such, when two atoms are excited to Rydberg states, they exhibit a strong van der Waals interaction. Placing N atoms at positions $\mathbf{r}_j$ in a 2D plane, and coupling the ground state $|0\rangle$ to the Rydberg state $|1\rangle$ with a laser system enables the realization of the Hamiltonian :

$$H = \sum_{i=1}^{N} \frac{\hbar\Omega}{2} \sigma_i^x - \sum_{i=1}^{N} \frac{\hbar\delta}{2} \sigma_i^z + \sum_{j<i} \frac{C_6}{|\mathbf{r}_i - \mathbf{r}_j|^6} n_i n_j. \tag{6}$$

Here, Ω and δ are respectively the Rabi frequency and detuning of the laser system and $\hbar$ is the reduced Planck constant. The first two terms of Eq. (6) govern the transition between states $|0\rangle$ and $|1\rangle$ induced by the laser, while the third term represents the repulsive Van der Waals interaction between atoms in the $|1\rangle$ state. More precisely, $n_i = (\sigma_i^z + 1)/2$ counts the number of Rydberg excitations at

position i. The interaction strength between two atoms decays as $|\mathbf{r}_i - \mathbf{r}_j|^{-6}$.

The shift in energy originating from the presence of two nearby excited atoms induces the so-called *Rydberg blockade* phenomenon, illustrated in Fig. 4(a). More precisely, if two atoms are separated by a distance smaller than the Rydberg blockade radius $r_b = (C_6/\hbar\Omega)^{1/6}$, the repulsive interaction will prevent them from being excited at the same time. On the other hand, the sharp decay of the interaction allows us to neglect this interaction term for atoms distant of more than r_b. As such, for $\Omega = 0$, the Hamiltonian in Eq. (6) is diagonal in the computational basis and enables to realize $H|z_1, \ldots, z_N\rangle = (\hbar\delta/2)C(z_1, \ldots, z_N)|z_1, \ldots, z_N\rangle$, with the cost function specified in Eq. (5), and for which there is a link between atoms i and j if they are closer than r_b apart.
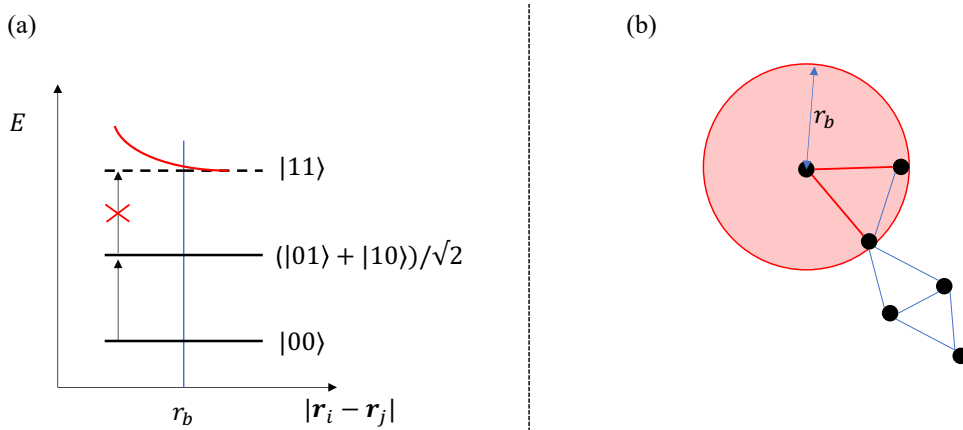

Figure 4: (a) Illustration of the Rydberg blockade effect. When two atoms are far apart, $|\mathbf{r}_i - \mathbf{r}_j| > r_b$, they don't interact. On the other hand, if they are separated by less than the Rydberg blockade radius, $|\mathbf{r}_i - \mathbf{r}_j| < r_b$, a strong interaction prevents the two atoms to be in the state $|1\rangle$ at the same time. (b) Rydberg blockade and independent sets of a graph. Rydberg atoms correspond to the nodes of a UD-graph. There are edges between adjacent nodes if the distance between them is smaller than the Rydberg blockade radius r_b, as illustrated for the top-left node. Due to the Rydberg blockade effect, the dynamics of the system is restricted to independent sets.

We have seen that atom arrays enable to study UD-MIS problems with QAOA using an analog control, which will likely offer better performances as compared to a digital approach. In the following, we present a way to transform interval graphs into UD graphs.

3.3.2 From Interval Graphs to Unit Disk Graphs

Graphs such as the ones used to model our Job Interval Selection problem (SC2) do not correspond to UD graphs, as they are unions of *interval intersection graphs* (the edges corresponding to the overlapping tasks) and of *cluster graphs* (the set of complete disjoint cliques corresponding to the groups) and thus *one-dimensional* intersection graphs, rather than *two-dimensional* ones. Hence, we have to transform our scheduling graphs in order to implement the search of their MIS on the quantum machine.

The correspondence between the original problem graph and the locations of the Rydberg atom/qubits

in the machine can be obtained by solving the following continuous quadratic problem (UD):

$$Min_{x_q,y_q}\{L\} \tag{7}$$

$$s.t.$$

$$\forall \langle i,j \rangle \in E, \rho^2 \leq (x_j - x_i)^2 + (y_j - y_i)^2 \leq r^2 \tag{8}$$

$$\forall \langle i,j \rangle \in \overline{E}, (x_j - x_i)^2 + (y_j - y_i)^2 > r^2 \tag{9}$$

$$\forall i \in V, x_i < L; y_i < L \tag{10}$$

$$L \leq \overline{L} \tag{11}$$

where:

- $G = (V, E)$ is the original interval graph of the problem;

- $\overline{G} = (V, \overline{E})$ is G's complementary, i.e. the graph whose vertices (i, j) are connected iff they are not connected in G;

- r is the Rydberg blockade radius, i.e. the upper bound on the distance between two connected vertices, $\rho \sim r/3$ a given factor defining the lower bound on their distance, illustrating the minimal spacing between separated atoms that is experimentally feasible;

- (x_i, y_i) are the coordinates in the euclidean plan of the atom/qubit representing the node $i \in V$;

- L is the minimum length of the side of the square required to embedded the graph while enforcing constraints above;

- $\overline{L}$ defines the maximal surface available to place the atoms on the machine.

Experimentally, the Rydberg blockade radius r can be around 15μm, for minimal distances between atoms of the order of 5μm, and the atoms are contained in a region characterized by $\overline{L} \sim 100\mu$m.

Due to the lower bound in constraints (8) and to constraints (9), this problem is not convex, and is thus NP-hard in the general case [44–46]. Furthermore, L clearly depends on the machine characteristics $(r, \rho, \overline{L})$, and on the structure of the original interval graph G, and the (UD) problem does not necessarily have feasible solutions. For example, it is clear that for poorly connected graphs, i.e with few load intervals overlapping or many groups with a low k, the requirement to keep away from each other atoms corresponding to non-connected nodes might render the problem infeasible in the square $\overline{L} \times \overline{L}$. On the contrary, strongly connected graphs should make difficult to satisfy the constraint of bringing together the atoms corresponding to connected nodes in the rings delimited by (r, ρ). We have tested different methods to solve the (UD) problem, allowing to produce feasible locations of Rydberg qubits on the quantum processor for various real smart-charging graphs in a short elapsed computation time. They are presented in Appendix A.

However it is worth noting that we had to solve the (UD) problem only for graph instances whose vertices's number is limited. It is clear that tackling larger load scheduling problems on larger quantum processors with this method could turn out to be difficult, due to the cost of computing the corresponding "UnitDisk" graph of Rydberg atoms. Overcoming this mapping issue is an important research objective that we leave for future work.

4 Results

4.1 Analytical result for Max-Cut: the mean cost in the $QAOA_1$ state

An analytical expression for Max-Cut on unweighted graphs for $p = 1$ was derived in [47]. We extend this result to the case of complete weighted graphs. For an edge $\langle u, v \rangle$ the mean value of $C_{u,v}$ for chosen γ, β is:

$$
\langle \mathbf{z}_{\gamma,\beta} | C_{u,v} | \mathbf{z}_{\gamma,\beta} \rangle = \frac{w_{uv}}{2} \left(1 + \frac{\sin(4\beta)\sin(\gamma w_{uv})}{2} \left[\prod_{x \neq u,v} \cos(\gamma w_{ux}) + \prod_{x \neq u,v} \cos(\gamma w_{vx}) \right] + \right.
$$
$$
\left. - \frac{\sin^2(2\beta)}{2} \left[\prod_{x \neq u,v} \cos(\gamma(w_{ux} - w_{vx})) - \prod_{x \neq u,v} \cos(\gamma(w_{ux} + w_{vx})) \right] \right) \quad (12)
$$

This is an important result for numerical experiments as the obtained expression allows to compute the mean cost in the $QAOA_1$ state without any call to a (simulated) quantum computer in polynomial time, which in turn is useful to analyze the performance of QAOA.

In this expression, we realize that there is a close link between the value of the QAOA angle γ and the weight of the edge w_{uv}. Discussions on the normalization of the graph edges and its impact on the energy landscape are covered in Appendix B.

4.2 Numerical results

In the following, we present a performance analysis of the various procedures presented above. For Max-Cut at depth $p = 1$, we use the analytical formula (12). For Max-4-Cut, MIS, and Max-Cut at depth $p > 1$, we compute the mean cost by Monte-Carlo estimation, by simulating the quantum evolution either on the Atos Quantum Learning Machine [48] or using the QuTIP library [49] on the OCCIGEN supercomputer based in Montpellier, France.

4.2.1 Real Data Set Used

Data were driven from a set of 2250 loads performed during May 2017 on identical charging points of the Belib's network of load stations located in Paris, France [50].

For both problems (SC1) and (SC2), an instance is a series of chronological loads characterized by their duration for (SC1) and starting/end times for (SC2). The instance size is the number of loads it contains. A data-set is a set of instances whose first load is randomly chosen among the 2250, according to a uniform law. Once an instance is built:

- for (SC1), a priority is randomly affected to each load, according to a Poisson's law, enforcing a constant distribution of the different priority levels in each instance[4];

- for (SC2), the belonging to a group is randomly affected to each load according two a uniform law parameterized by the number of groups and the number of loads in the instance.

---

[4]The "brute" weights from the data yield high values in the cost Hamiltonian, leading to a highly fluctuating energy landscape for $QAOA_1$ (cf. equation 12). A re-weighting was therefore done in order to obtain a smoother landscape for better optimization. (see Appendix B).

13

4.2.2   Minimization of Total Weighted Load Completion Time (SC1)

First, we compare the performance of QAOA and the randomized algorithm on Max-Cut instances of different sizes $N \in [6, 8, 10, 15, 30, 50, 70, 100, 150]$. Both QAOA and the randomized algorithm return a sample from a certain probability distribution (built by a quantum circuit and uniform distributions respectively). In order to compute the exact optimum C_{opt} we use brute-force search for Max-Cut on small instances (with up to 30 nodes) and the dynamic program algorithm presented in Ref. [51] for the initial scheduling problem on bigger instances.
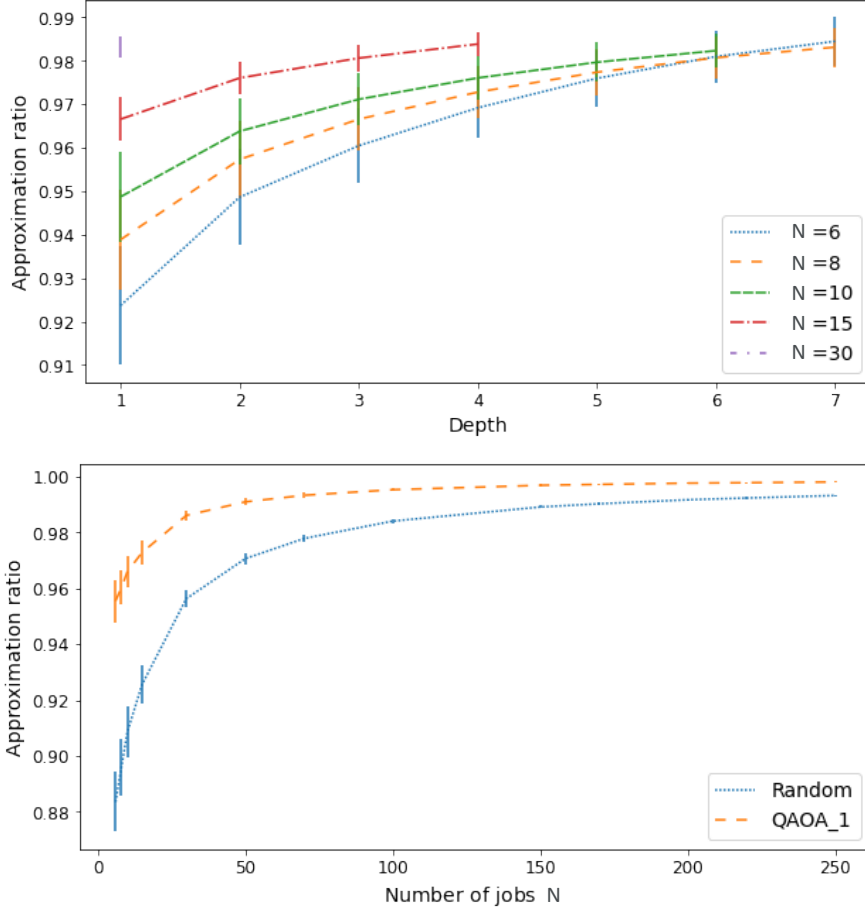
Figure 5: Top panel: Evolution of the approximation ratio of $QAOA$ with depth p for the Max-Cut problem. Bottom panel: Evolution of the average approximation ratio with the instance size for $QAOA$ at depth $p = 1$ (dashed orange line) and for the randomized algorithm on the *initial scheduling problem* (dotted blue line).

As expected, we observe on the top panel of Fig. 5 that, for a fixed value of N, the approximation ratio improves with the QAOA depth p.

Surprisingly, we also notice that the approximation ratio gets better with the size of the Max-Cut instance. Such behavior is also observed for the randomized algorithm, as illustrated on the bottom panel of Figure 5. In this numerical experiment we observe that QAOA finds better solutions than the randomized algorithm.

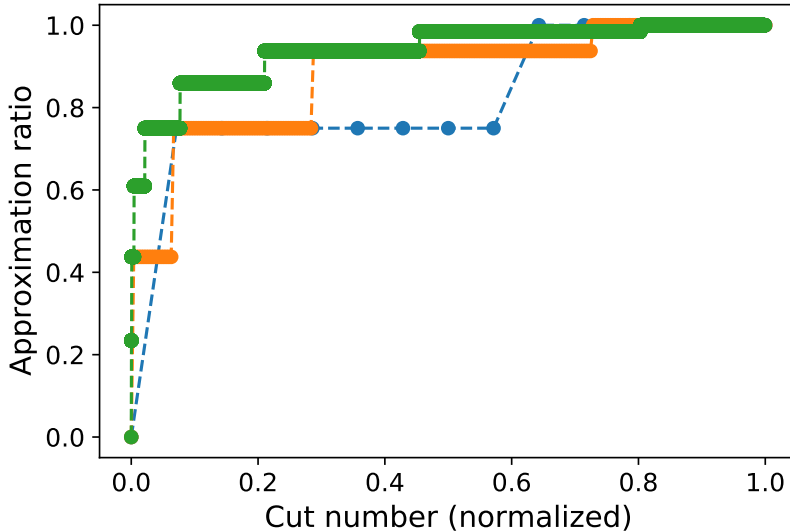The good performances of the randomized algorithm shown on the bottom panel of Figure 5 sug-

14

Figure 6: Approximation ratios of all possible cuts in the unweighted Max-Cut problem on complete graphs of size N, sorted by increasing values. The blue, orange and green dots correspond to $N = 4$, $N = 8$, and $N = 16$, respectively.

gest that the difficulty of the problem under consideration decreases with its size. To confirm this insight, let us consider the simpler case of unweighted Max-Cut on a complete graph. In this scenario, choosing a cut at random gives an approximation ratio of $\left[\sum_k \binom{N}{k} k(N - k)\right] / (N^2/4)$ which goes to 1 in the large N limit. This fact is illustrated in Fig. 6, where we show the approximation ratios of all possible cuts for different values of N. This plot shows that, for a fixed positive value of the normalized cut number, the corresponding approximation ratio approaches 1 in the large N limit. While our findings suggest that the presence of $\mathcal{O}(1)$ weights in our problems leads to the same behavior, it does not extend to instances in which the magnitude of the weights would increase with N.

We now present numerical results for Max-4-Cut. Using the normalization factor introduced in Appendix B, we ran QAOA for $p = 7$ layers, and plotted the statistical distribution of the approximation ratio achieved for 98 different instances (Fig. 7). As expected, the approximation ratio increases to 1 with the number of layers. The high value of the approximation ratio achieved is an encouraging result, showing that quantum approaches are comfortably higher than the classical minimal approximation guarantee of 0.857487 [52], even at low depths. At $p = 1$, the tail of the distribution indicates that some instances have been poorly optimized. The fact however that this tail disappears in the next layers shows that initial poor optimization can be corrected in the following layers. This is due to the fact that at the end of each layer our classical algorithm implements a rapid local optimization on all parameters.

4.2.3 Optimal Scheduling of Load Time Intervals within Groups (SC2)

Previous studies on quantum approaches for solving UD-MIS investigated the influence of quantum noise QAOA [43], and compared the performances of Quantum Annealing procedures [53] to classical
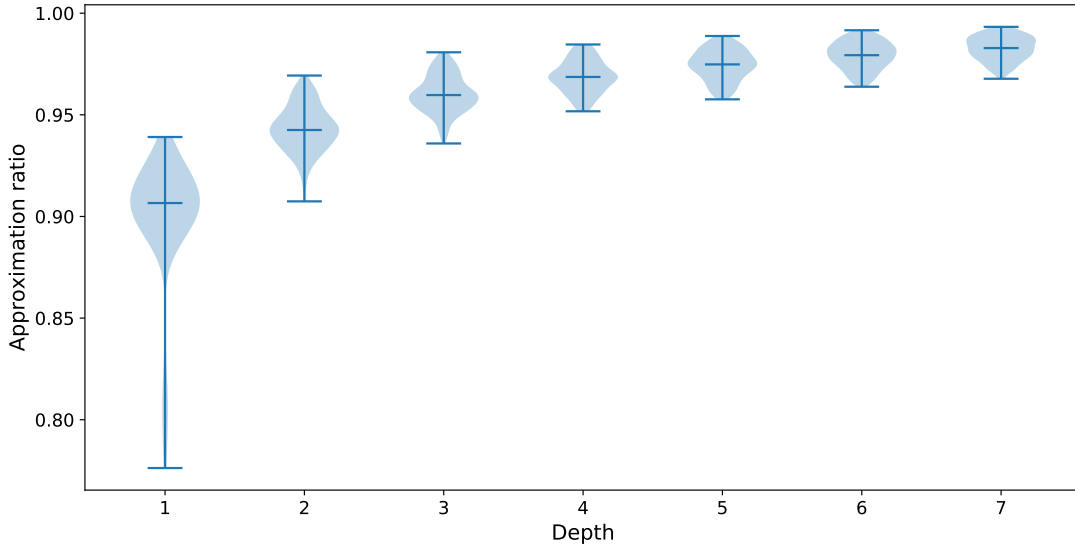
Figure 7: **Max-4-Cut**. QAOA for $p = 7$ layers, and the statistical distribution of the approximation ratio achieved for 98 instances. Proper re-weighting of the graph uniforms the instances. The high value of the approximation ratio achieved is an encouraging result, showing that quantum approaches are comfortably higher than the classical approximation minimal guarantee of 0.857487 [52], even at low depths. While initial optimization might not be ideal, as suggested by the tail of distribution on the first layer, it is corrected by the global smoothing at the next layer.

approximation algorithms[5].

For the classical loop of QAOA, we use here a new global optimization process (Egg optimization) described in the Appendix C.1 that uses a global optimization process called differential evolution (DE) [54]. Using (DE) enables to escape from local minima quite easily. Another important advantage is that the function evaluations can be done in parallel. More specifically, we ran our program on the OCCIGEN supercomputer based in Montpellier, France, where we used 28 CPU cores in parallel for our calculations. This reduced by a factor of 8.8 the time of derivation. In addition, we reduced the amount of phase space addressed by making educated guesses from layer $p - 1$ to layer p, strongly inspired by Ref. [55], and demonstrated precise results with much fewer function calls. The combination of educated guesses and parallel function evaluations made for a consequent speed-up, bringing down typical calculation times of half a day to an hour. Finally, the global optimizer (DE) has shown strong results in the context of noisy and changing landscape [56], a typical behavior of our noisy intermediate-scale quantum platforms. Using (DE) might prove robust in an experimental setup.

As can be seen in Fig. 8(a), the performances of QAOA for solving the UD-MIS problem are good in average, exceeding approximation ratios of 0.95 after seven QAOA layers. As can be noted in the figure, the distribution of the approximation ratio for each layer is rather wide. At the third layer, the distribution starts to separate in two bulks. The lower bulk stagnates by the fourth layer as the approximation ratio stays inferior to 0.85 until the last layer, while the approximation of the upper bulk increases to one as the depth grows. For completeness, we show on panel (b) and (c) of Fig. 8 a typical
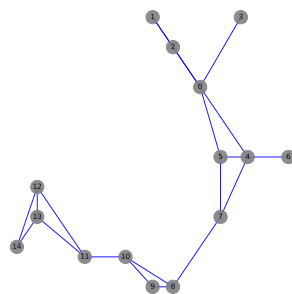
---

[5]Note that, for the small instances studied here, classical brute-force algorithm can find the exact solution in a short time

16
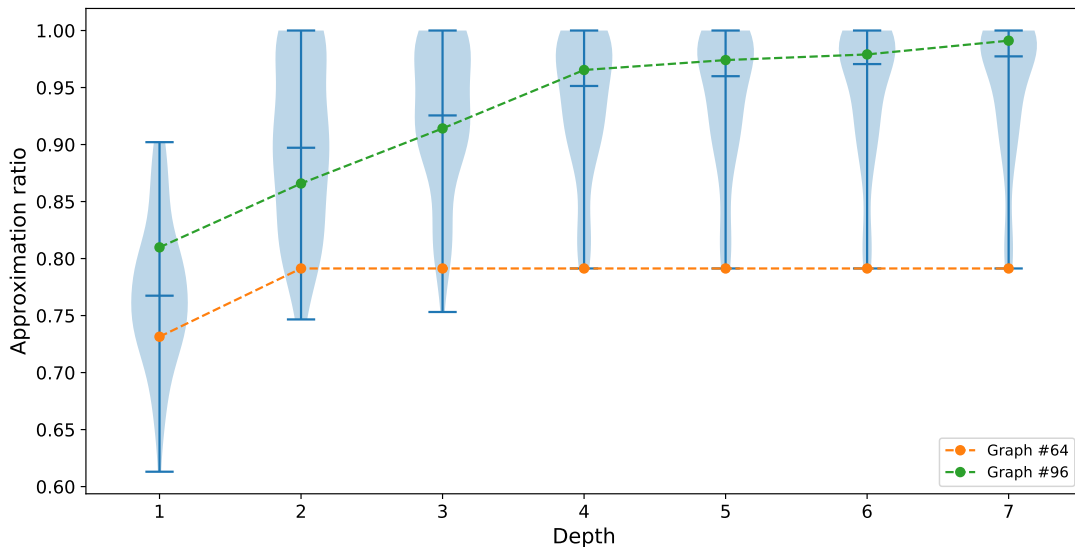
(a) Evolution of the approximation ratio for 84 instances of Unit-Disk MIS ($N = 15$).
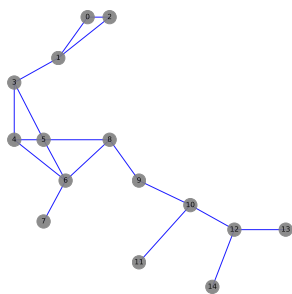
(b) Graph #96.

(c) Graph #64.

Figure 8: **(a)** For each layer, the median is plotted and the violin envelop represents the distribution of the approximation ratio achieved for each of the 84 points. While the approximation ratio grows rapidly to 1 in most cases, there seems to be some instances that are harder to optimize. We isolate a graph representative of the best-case bulk **(b)** and one for the worst-case bulk (**c**). Further understanding of the worst-case graphs is important in order to appreciate the quality of a quantum approximation algorithm.

graph instance of each group. The instances in the lower bulk correspond to worst-case scenarios and represent 9.5% of the instances. Understanding the characteristics of the worst instances is of crucial importance to characterize the quantum approaches. Indeed in the approximation theory the quality of an algorithm is benchmarked on worst-case instances. The approximation ratio achieved by the algorithm on these particular instances is a guarantee from below for any other instance. Finding worst-case scenario has been investigated in the past for Max-Cut on uniform 3-regular graphs at depths $p = \{1, 2, 3\}$ [11, 57]. Obtaining a lower bound guarantee of QAOA on UD-MIS, which we leave for future work, would enable us to compare the quantum approach to the classical approximation scheme to assess an eventual quantum advantage.

5   Conclusions and Perspectives

Qualifying quantum algorithms on difficult optimization problems is of great importance to evaluate the benefit of quantum computing, as these problems are at the core of many industrial applications where they often constitute performance bottlenecks.

Two major principles must be implemented in such a process:

1. Rely on a collaboration between experts in the application field under study and quantum computing experts, in order to design fine-tuned, *ad-hoc*, software-hardware solutions;

2. Benchmark quantum algorithms not only against exact classical algorithms, by nature exponential on this class of problems, but also versus available approximate polynomial ones.

The first point is crucial, because quantum solutions can often take advantage of the specific characteristics of the targeted quantum hardware. The second point is required for a fair-play competition between quantum and classical approaches for difficult optimization, in order to precisely evaluate a potential quantum advantage.

This paper reports a case study based on this protocol in the field of smart-charging of electric vehicles. We specified two smart-charging problems, which, although stylized to be treated by available quantum approaches, stay representative of the real operational problems currently solved by the EDF subsidiaries involved in the field. We developed a hardware-efficient implementation of QAOA on quantum devices based on Rydberg atoms arrays to solve these two problems, respectively modeled as "sub-difficult" instances of Max-k-Cut and MIS NP-hard problems. We have experimented these implementations on a real data-set of 2250 loads, and compared quantum solutions to classical approximate ones, up to current limits of classical simulation of quantum hardware with $N \le 20$ qubits. In both cases, quantum algorithms behave correctly, obtaining high approximation ratios, coherently with the fact that both applications are modeled as "less difficult" instances than the worst-case ones of these two NP-Complete problems.

These results, obtained through a rigorous protocol, are very encouraging. Future works will involve testing the quantum approaches on the real Rydberg atoms quantum processor developed by Pasqal in the 100-1000 qubits range [16]; making the smart-charging problems more realistic by incorporating new constraints (e.g maximal available power on the load station), a real challenge as this should make the associated Hamiltonians to be implemented on the processor more complex; more specifically from an application viewpoint, looking for efficient heuristics to transform general graphs in unit-disk ones, which would drastically simplify the procedure for quantum solving of MIS. On this latter point, another interesting option is to explore smart-charging problems which are "naturally" two dimensional – e.g. based on the "autonomy radius" of vehicles or the "action radius" of charging points –, thus replacing the costly and hypothetical resolution of the (UD) problem by a simple scale reduction in the plan.

Acknowledgments

A Mapping interval graphs to unit-disk graphs of Rydberg atoms on the quantum machine

The (UD) problem introduced section 3.3.2 can be solved by a classical reformulation-linearization of non-convex quadratic programs into binary/integer ones, by considering variables x_i and y_i as integers and replacing them by their binary expansions in constraints(8) and (9), then adding binary variables and appropriate constraints to represent the products between them [44–46]. The resulting binary/integer linear model is as follows:

$$Min_{x_q,y_q}\{L\} \tag{13}$$

$$s.t.$$

$$\forall i \in V, x_i = \sum_{k=0}^{k=log(\overline{L})} 2^k bx_i^k; \quad y_i = \sum_{k=0}^{k=log(\overline{L})} 2^k by_i^k \tag{14}$$

$$\forall i \in V, x_i < L; y_i < L \tag{15}$$

$$L \leq \overline{L} \tag{16}$$

$$\tag{17}$$

$$\forall \langle i,j \rangle \in E : \tag{18}$$

$$\rho^2 \leq X_j X_j - 2X_i X_j + X_i X_i + Y_j Y_j - 2Y_i Y_j + Y_i Y_i \leq r^2 \tag{19}$$

$$X_i X_j = \sum_{k=0}^{k=log(\overline{L})} \sum_{k'=0}^{k'=log(\overline{L})} 2^{(k+k')} wx_{i,j}^{k,k'} \tag{20}$$

$$Y_i Y_j = \sum_{k=0}^{k=log(\overline{L})} \sum_{k'=0}^{k'=log(\overline{L})} 2^{(k+k')} wy_{i,j}^{k,k'} \tag{21}$$

$$wx_{i,j}^{k,k'} \leq bx_i^k; \quad wx_{i,j}^{k,k'} \leq bx_j^{k'}; \quad wx_{i,j}^{k,k'} \geq bx_i^k + bx_j^{k'} - 1 \tag{22}$$

$$(UDRLT) \quad wy_{i,j}^{k,k'} \leq by_i^k; \quad wy_{i,j}^{k,k'} \leq by_j^{k'}; \quad wy_{i,j}^{k,k'} \geq by_i^k + by_j^{k'} - 1 \tag{23}$$

$$\tag{24}$$

$$\forall \langle i,j \rangle \in \overline{E} : \tag{25}$$

$$X_j X_j - 2X_i X_j + X_i X_i + Y_j Y_j - 2Y_i Y_j + Y_i Y_i > r^2 \tag{26}$$

$$X_i X_j = \sum_{k=0}^{k=log(\overline{L})} \sum_{k'=0}^{k'=log(\overline{L})} 2^{(k+k')} wx_{i,j}^{k,k'} \tag{27}$$

$$Y_i Y_j = \sum_{k=0}^{k=log(\overline{L})} \sum_{k'=0}^{k'=log(\overline{L})} 2^{(k+k')} wy_{i,j}^{k,k'} \tag{28}$$

$$wx_{i,j}^{k,k'} \leq bx_i^k; \quad wx_{i,j}^{k,k'} \leq bx_j^{k'}; \quad wx_{i,j}^{k,k'} \geq bx_i^k + bx_j^{k'} - 1 \tag{29}$$

$$wy_{i,j}^{k,k'} \leq by_i^k; \quad wy_{i,j}^{k,k'} \leq by_j^{k'}; \quad wy_{i,j}^{k,k'} \geq by_i^k + by_j^{k'} - 1 \tag{30}$$

$$bx_i^k, by_i^k, wx_{i,j}^{k,k'}, wy_{i,j}^{k,k'} \text{all binary.} \tag{31}$$

Once rewritten this way, and assuming that it is feasible, (UD) can be solved to optimality by conventional Branch and Bound/Cut algorithms, but with an exponential computing time in the number of integer/binary variables, in the worst case. This renders this formulation too costly even for

small sized graphs, as for each couple (x_i, y_i) of continuous variables modeling a vertex in the original problem we introduce $2log(\overline{L})$ binary variables (bx_i, by_i) to expand its integer formulation, and $(log(\overline{L}))^2$ binary variables $wx_{i,j}^{k,k'}/wy_{i,j}^{k,k'}$ to express each product in the constraints related to the edges.

As an alternative, we used a more compact linear formulation which, although not providing guaranty to find an existing solution in the elapsed time allowed, did perform well on the majority of smart-charging graphs tested. Basically, the idea is to linearize the problem by replacing the quadratic constraints by linear ones, expressing the belonging to the rings defined by (r, ρ) by means of a given set of parametrized radius. The resulting model is as follows:

$$Min_{x_q, y_q}\{L\} \tag{32}$$

$$s.t.$$

$$\forall i \in V, x_i < L; y_i < L \tag{33}$$

$$L \leq \overline{L} \tag{34}$$

$$(UDR_\Phi) \qquad \forall\langle i, j\rangle \in E : \bigvee_{\phi \in \Phi} f_\phi(x_i, y_i, x_j, y_j) \tag{35}$$

$$.\forall\langle i, j\rangle \in \overline{E} : (|x_j - x_i| > r) \vee (|y_j - y_i| > r) \tag{36}$$

where f_ϕ is a set of clauses enforcing the belonging of the point (x_j, y_j) to the radius defined by the angle ϕ intersecting the ring of center (x_i, y_i) defined by (r, ρ). For example:

$$f_{\phi \in [0, \frac{\pi}{2}[} = (\rho \cos \phi \leq x_j - x_i) \wedge (x_j - x_i \leq r \cos \phi) \wedge (\rho \sin \phi \leq y_j - y_i) \wedge (y_j - y_i \leq r sin\phi)$$

Because of the "or" and "absolute value" terms in the formulation, this model is still combinatorial, but much more efficient than the previous one.

It is worth noting that in both formulations, it is possible to relax the constraint expressed in the objective function enforcing to locate the Rydberg atoms in the smallest square contained in $\overline{L} \times \overline{L}$; one thus obtains a pure "constraint programming" problem, whose solving can stop at the first solution encountered.

Both formulations were implemented and tested with Ibm Cplex solver. The second one provided in one hour of elapsed time with 57 instances to be implemented for UD-MIS search on the Rydberg atom based quantum machine, starting from 100 instances of real smart-charging graphs of 15 nodes each.

B Optimizing the classical loop of QAOA

Although translation of a combinatorial problem focuses on the quantum loop of the QAOA, the quality of the classical optimizer part is of crucial importance. Indeed, an inefficient classical optimizer will bring excessive overhead to the whole process. For practical use, it is of crucial importance to use an optimizer tailored to the machine. In the following paragraphs, we present results on the research for the best classical optimization process.

In the (SC1) problem, the weights w_{uv} of the edges in the graph are imposed by real data. These weights appear in eq.(12) and multiply the parameterized angle γ. They therefore impact the phase that is applied to each basis state of the computational basis in the QAOA cycle. To visualize the impact of weights, we plot the energy (or cost-function) landscape at level $p = 1$ as a function of the parameters (γ, β). Fig.9 contains numerical simulation of the energy landscape for the Max-Cut
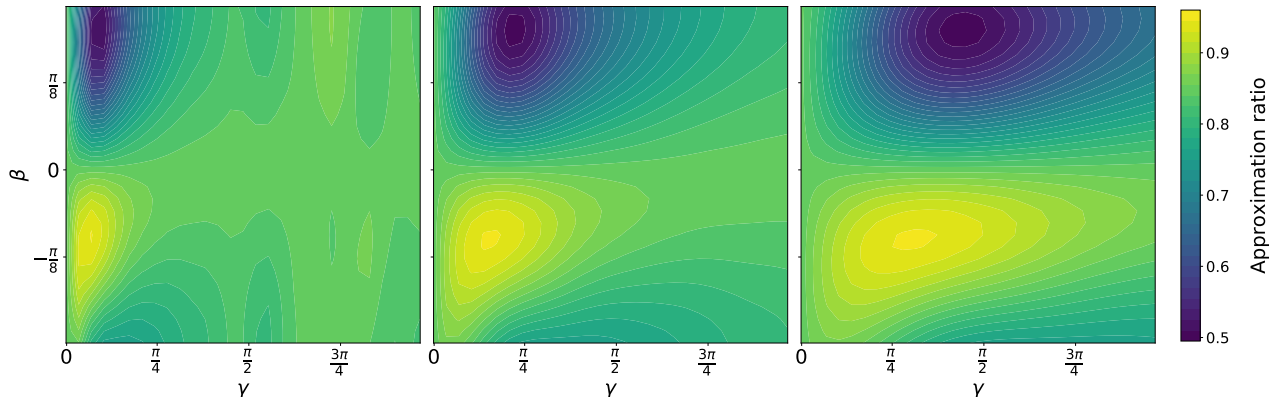
Figure 9: **The zoom effect**. From left to right, we normalized an instance from (SC1) respectively by $R/5$, $R/2$ and R to observe the effect on the energy landscape of $QAOA_1$. The closer to R, the better the zoom on the global minima. The re-weighting of a graph affects the energy landscape: it can therefore be used as leverage to either zoom on the point of interest to apply local optimization, or on the contrary it might be used to zoom out of barren plateaus to explore more interesting phase spaces.

problem, evaluated on a graph of size 10, with a resolution of 30 points along each axis. For the same problem, we re-weight the adjacency matrix of the graph by different factors. On the first figure, a high density of peaked valleys and hills indicates that optimization is difficult and would require an important amount of function evaluations to find a decent solution. The amount of peaks and valleys is due to the fact that the cost Hamiltonian adds a phase term to each basis vector z_i. Indeed, applying $\hat{C}$ with an angle γ modifies $|z_i\rangle$ to $e^{i\gamma C_i}|z_i\rangle$, where C_i corresponds to the cost of the coloring z_i. Modifying the graph weights consequently modifies the the phase applied to each basis state z_i. A smaller C_i, as seen in the third figure, smooths the cost-function landscape enabling adequate local optimization. Artificially reducing C_i too much however might over-smooth the landscape, reducing the possible phases that QAOA can apply, hence missing the global minima. We want C_i to be big enough to allow basis states z_i to acquire a phase $e^{i\phi}$ in a comfortable range. At the same time, we do not believe a phase $\phi > 2\pi$ is necessary. The re-weighting $C_i = R^* C_i$, where $R^* = \frac{2\pi}{\max(C_i)}$ satisfies the two previous conditions. Numerically, we find that re-weighting the adjacency matrices of all instances by the factor R indeed concentrates all optimal parameters in a restricted zone of parameter space.

Practically of course, calculating R^* implies knowledge of $\max C_i$, which corresponds to the exact solution to the initial problem. We therefore propose the upper bound $R = w_{\max} \frac{N^2}{4}$, where $w_{\max} = \max_{u,v} w_{uv}$. It is calculated from the best-cut on a complete graph where all weights would be equal to $w_{\max}$.

In the scenario of (SC1), we see that the choice for the optimizer depends on the normalization of the weights. If one manages a good normalization, then local methods of optimization can be used with high guarantees of success. In general, we advocate the use of a global optimizer if there is no prior knowledge of the energy landscape.

C Finding correct parameters: the Egg optimization and local vs. global methods

C.1 Educated global guess (Egg) optimization

In order to find the best variational parameters for p layers, we develop a method based on the idea of making an educated guess from previous layers to the new one [55]. The educated global guess (Egg) optimization process uses the differential evolution (DE) [54] rather than a local optimization in an attempt to find the global optima in a wrinkled energy landscape. (DE) works by starting with an ensemble of points in the phase-space, called the *agent population*. Then, theses agents are moved around by recombining their coordinates, and the function is evaluated for these new agents. If the new position brings an improvement, it is kept, otherwise it is discarded. This process is repeated until convergence to a minima, although there is no guarantee that the global minima will be found. While it cannot be sure that our method will always work perfectly, the constant growth of the approximation ratio as p increases in our results is a reassuring indicator, as illustrated on the top panel of Fig. 8. A second major hurdle in the optimization process is the high-dimensionality of the phase space: for p layers, we need to find the global minimum of a space of size $2p$. We strongly reduced the complexity of the problem by making a global educated guess for the optimal parameters at layer p using the optimal parameters found at layer $p-1$ (Alg.1). The heuristic of an educated guess from the previous layer to the new one was developed in Ref. [55]. Our version uses a global optimizer rather than a local one in an attempt to find the global optima in a wrinkled energy landscape. This very much improved computation time by reducing the amount of phase space addressed, and demonstrated precise results with much fewer function calls than local methods (see Appendix C). The algorithm is described in pseud-code below (Alg.1). It works as follows: find optimal parameters (γ_1^*, β_1^*) for $p = 1$. For the next layer, optimize the function $C : (\gamma_2, \beta_2) \mapsto C(\gamma_1^*, \gamma_2, \beta_1^*, \beta_2)$. As such, two variables are already fixed and the space to explore is once again only bi-dimensional. Once the optimization ends on the two new coordinates, a local optimization is done on all the coordinates. This quick step enables to re-calibrate the previous parameters: it is therefore possible to achieve a trotterization process for high values of p.

Algorithm 1: EDUCATED GLOBAL GUESS (EGG) OPTIMIZATION

Input : $|z_0\rangle, \hat{C}, \hat{M}, p$.
Output: Optimal parameters $(\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)$.

---

1 $\gamma_1, \beta_1 \leftarrow \text{DE}(|z_0\rangle, \hat{C}, \hat{M})$
2 for k *in range*$(2, p)$ **do**
3 $\quad |z_{k-1}\rangle = e^{i\beta_{k-1}\hat{M}} e^{i\gamma_{k-1}\hat{C}} \ldots e^{i\beta_1\hat{M}} e^{i\gamma_1\hat{C}} |z_0\rangle$
4 $\quad \gamma_k, \beta_k = \text{DE}(|z_{k-1}\rangle, \hat{C}, \hat{M})$
5 $\quad$ **local optimization**: $\text{BFGS}(\gamma_1, \ldots, \gamma_k, \beta_1, \ldots, \beta_k, |z_0\rangle, \hat{C}, \hat{M})$
6 end
7 return $(\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)$


C.2 Local vs. global methods

In Ref. [10] the grid search was proposed to find optimal parameters for QAOA. This method returns a global optima but quickly becomes intractable while growing the depth p. Thus a practical approach is to use numerical optimization methods. They may be divided in two groups: global and local methods. Global optimization procedures (such as grid search) address a higher amount of the search space while

the local ones return a solution that lays in a neighborhood of a predefined initial point.

An experimental protocol for QAOA should specify the method to use in the parameter optimization step as well as its additional parameters. Local optimization routines are often used in works on applied QAOA [58–60], however the reasons why a certain method is chosen for a particular application are usually omitted. For (SC1) we compared different methods in terms of their performances (evaluated by the approximation ratio of the QAOA with returned parameters) and costs (measured in number of calls of the function to optimize, i.e. a function that computes $\langle z_{\gamma,\beta} | \hat{C} | z_{\gamma,\beta} \rangle$).

In order to apply a local optimization routine one has to specify a good initialization point. Without any prior knowledge about the parameter space it is chosen randomly from a set of possible values (usually several points are studied in parallel). However in some applications a pattern in optimal parameters is observed. Such pattern may rely optimal parameter values at different depths, as in [55], or the values for certain groups of instances [61].
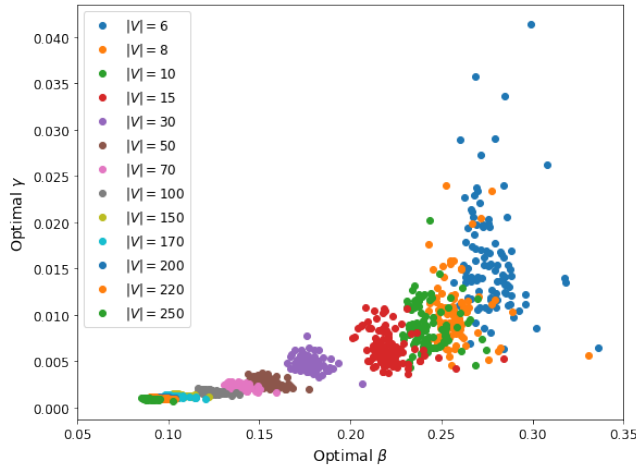
(a) $QAOA_1$ optimal parameters for instances of different sizes
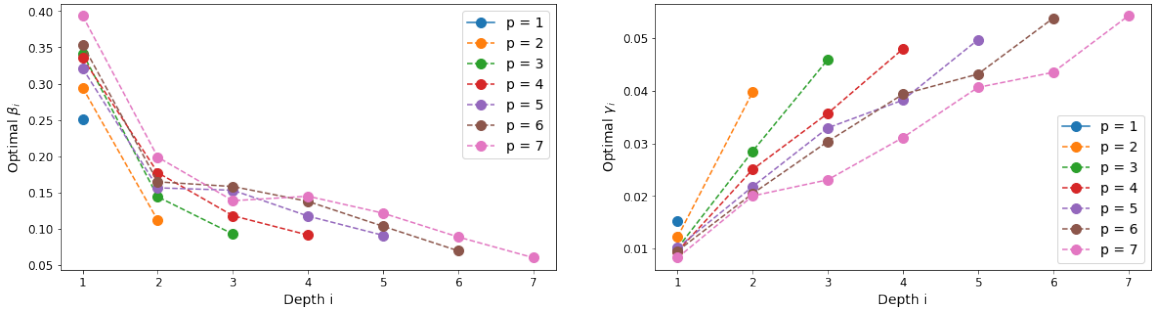
(b) An example of optimal parameters for different p for an instance with $|V| = 8$

Figure 10: Initialization strategy for numerical optimization: a pattern is observed for $p = 1$ (a) and the a linear transformation of parameters at $p - 1$ for $p > 1$ (b)

For $QAOA_1$ the equality $\gamma = 0$ implies the absence of any quantum effect (the resulting distribution stays uniform). In Fig. 10 we observe that the absolute values of β and γ get smaller for instances with bigger $|V|$, but in general it does not imply that the quantum effect becomes less important as γ may changed to an arbitrary positive value by multiplying all weights with some factor. We also can't claim that $QAOA_1$ output distribution gets closer to uniform even if we observed that their mean values effectively get closer.

24

We found that at depth $p = 1$ optimal parameter values are close for instances with the same number of nodes $|V|$ and that they get concentrated while growing the size of the graph 10. For $p = 1$ we developed an initialization procedure for local methods that relies on the observed pattern. At depths $p > 1$ we used the *INTERP* heuristic introduces in Ref. [55] which is based on the intuition that optimal parameters at depth p are close to a linear transformation of optimal values at $p - 1$, which seems to be true for our instances (see Fig. 10).
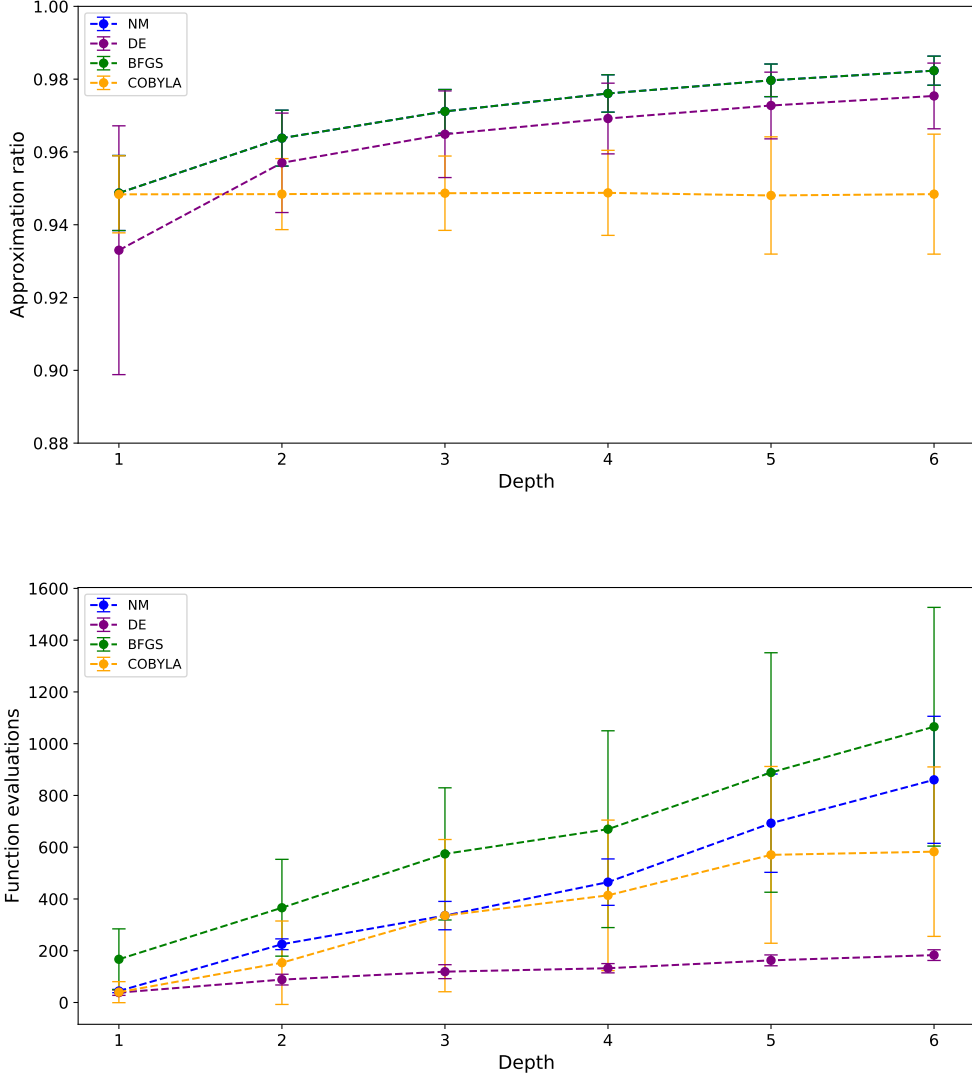

Figure 11: Cost and performance of different numerical optimization methods on Max-Cut (SC1), for graphs with size $N = 10$. Differential Evolution (DE) corresponds to a global optimization algorithm, inspired by the INTERP method. While yielding results very close to Nelder-Mead, it does so by requiring very little number of function evaluations. Building optimization processes such as DE that require little function evaluations is key in the NISQ era as it ensures quicker performances on unstable devices.

A choice of the optimization routine has a high impact on the performance and cost of an experimental implementation of QAOA as was shown in Ref. [62]. By experimentally comparing different

optimization methods we observed that for local methods, a gradient-free *Nelder-Mead* [63] is the best choice for our purposes: its performance is comparable to the one of the quasi-Newton *BFGS* [64] (and both methods outperform the *COBYLA* routine [65] which is often used in works on applied QAOA [58]) while it requires less evaluations of the objective function (see Fig. 11).

We compare these local methods with *Differential Evolution* (DE) [54]. We limited the number of function calls while using DE in order to have the best trade-off between global exploration and low number of function evaluation. As seen in Fig.11it is a very effective method, for which the approximation ratio grows close to Nelder-Mead using however much less function evaluations with the layers. It should be kept in mind nonetheless that global optimization methods require tuning hyper-parameters, a process that must be adjusted by hand. The optimal hyper-parameters might change from one problem to another, a reason why they are not so popular. But building optimization processes such as DE that require little function evaluations is key in the NISQ era as it ensures quicker performances on unstable devices.

References

1. Shor, P. W. *Algorithms for quantum computation: discrete logarithms and factoring* in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994), 124–134.

2. Grover, L. K. *A Fast Quantum Mechanical Algorithm for Database Search* in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, Philadelphia, Pennsylvania, USA, 1996), 212–219. ISBN: 0897917855. `https://doi.org/10.1145/237814.237866`.

3. Bennett, C. H., Bernstein, E., Brassard, G. & Vazirani, U. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing* **26,** 1510–1523. ISSN: 1095-7111. `http://dx.doi.org/10.1137/S0097539796300933` (Oct. 1997).

4. Viamontes, G. F., Markov, I. L. & Hayes, J. P. Is quantum search practical? *Computing in science & engineering* **7,** 62–70 (2005).

5. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2,** 79. ISSN: 2521-327X. `http://dx.doi.org/10.22331/q-2018-08-06-79` (Aug. 2018).

6. Choi, J.-y. *et al.* Exploring the many-body localization transition in two dimensions. *Science* **352,** 1547–1552. ISSN: 1095-9203. `http://dx.doi.org/10.1126/science.aaf8834` (June 2016).

7. Scholl, P. *et al.* Programmable quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms. *arXiv e-prints,* arXiv:2012.12268. arXiv: `2012.12268` [quant-ph] (Dec. 2020).

8. Kendon, V. Quantum computing using continuous-time evolution. *arXiv:2004.00704* (2020).

9. Satoshi, M. & Hidetoshi, N. Mathematical Foundation of Quantum Annealing. *arXiv:0806.1859* (2008).

10. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum Computation by Adiabatic Evolution. *arXiv:quant-ph/0001106.* (2000).

11. Farhi, E., Goldstone, J. & Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv:1411.4028* (2014).

12. Ibarra, O. H. & Kim, C. E. Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems. *J. ACM* **22,** 463–468. ISSN: 0004-5411. `https://doi.org/10.1145/321906.321909` (Oct. 1975).

13. Pichler, H., Sheng-Tao, W., Zhou, L., Choi, S. & Lukin, M. Quantum Optimization for Maximum Independent Set Using Rydberg Atom Arrays. *arXiv:1808.10816v1* (2018).

14. Saffman, M., Walker, T. G. & Mølmer, K. Quantum information with Rydberg atoms. *Reviews of Modern Physics* **82,** 2313–2363. arXiv: `0909.4777` [`quant-ph`] (July 2010).

15. Saffman, M. Quantum computing with atomic qubits and Rydberg interactions: progress and challenges. *Journal of Physics B Atomic Molecular Physics* **49,** 202001. arXiv: `1605.05207` [`quant-ph`] (Oct. 2016).

16. Henriet, L. *et al.* Quantum computing with neutral atoms. *Quantum* **4,** 327. ISSN: 2521-327X. `https://doi.org/10.22331/q-2020-09-21-327` (Sept. 2020).

17. Graham, R. L., Lawler, E., Lenstra, J. K. & Rinnooy Kan, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics* **5,** 287–326 (1979).

18. Skutella, M. *Semidefinite relaxations for parallel machine scheduling* in *39th Annual Symposium on Foundations of Computer Science Palo Alto, CA, USA* (1998), 472–481.

19. Yang, H. & Ye Yinyu. Zhang, J. An approximation algorithm for scheduling two parallel machines with capacity constraints. *Discrete Applied Mathematics* **130,** 449–467 (2003).

20. SKUTELLA, M. & WOEGINGER, G. J. *A PTAS for minimizing the total weighted completion time on identical parallel machines* CORE Discussion Papers 1999029 (Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), May 1999). `https://ideas.repec.org/p/cor/louvco/1999029.html`.

21. Woeginger, G. When Does a Dynamic Programming Formulation Guarantee the Existence of a Fully Polynomial Time Approximation Scheme (FPTAS)? *INFORMS Journal on Computing* **12,** 57–74 (Feb. 2000).

22. Smith, W. E. Various optimizers for single–stage production. *Naval Research and Logistics Quarterly* **3,** 59–66 (1956).

23. Goemans, M. X. & Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. of the ACM* **42(6),** 1115–1145 (1995).

24. Frieze, A. & Jerrum, M. Improved Approximation Algorithms for MAX k-CUT and MAX BI-SECTION. *Integer Programming and Combinatorial Optimation* **920,** 1–13 (1995).

25. De Klerk, E., Pasechnik, D. & Warners, J. On approximate graph colouring and max k-cut algorithms based on the ϑ function. *Journal of Combinatorial Optimization* **8(3),** 267–294 (2004).

26. Hastad, J. Some Optimal Inapproximality Results. *J. of the ACM* **48(4),** 798–859 (2001).

27. Kann, V., Khanna, S., Lagergren, J. & Panconesi, A. On the Hardness of Approximating Max-k-Cut and its Dual. *Chicago Journal of Theoretical Computer Science* **2,** 1–18 (1997).

28. Papadimitriou, C. H. & Yannakakis, M. Optimization, Approximation, and Complexity Classes. *J. of Computer and system sciences* **43(3),** 425–440 (1991).

29. Khot, S., Kindler, G., Mossel, E. & O'Donnell, R. Optimal Inapproximability Results for MAX-CUT and Other 2-variable CSPs? *SIAM Journal on Computing* **37(1),** 319–357 (2007).

30. Crooks, G. E. Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem. *arXiv e-prints,* arXiv:1811.08419. arXiv: `1811.08419` [`quant-ph`] (Nov. 2018).

31. Kolen, A. W., Lenstra, J. K., Papadimitriou, C. H. & Spieksma, F. C. *Interval Scheduling: A Survey* `www.interscience.wiley.com`. Wiley InterScience, 2007.

32. Spieksma, F. C. R. On the approximability of an interval scheduling problem. *J. of Scheduling* **2(5),** 215–227 (1999).

33. Chuzhoy, J., Ostrovsky, R. & Y., R. Approximation algorithms for the Job interval selection problem and related scheduling problemsInterval. *Mathematics of Operations Research* **31(4),** 730–738 (2006).

34. Bevern, R., Mnich, M., Niedermeier, R. & Mathias, W. Interval Scheduling and Colorful Independent Sets. *arXiv:1402.0851v2* (2014).

35. Chan, T. M. & Har-Peled, S. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete and Computational Geometry* **48(2),** 373–392 (2012).

36. Nieberg, T., Hurink, J. & K., W. *A Robust PTAS for Maximum WeightIndependent Sets in Unit Disk Graphs* in *International Workshop on Graph-Theoretic Concepts in Computer Science* (2004), 214, 221.

37. Cerezo, M. *et al.* Variational Quantum Algorithms. *arXiv e-prints,* arXiv:2012.09265. arXiv: `2012.09265 [quant-ph]` (Dec. 2020).

38. Hadfield, S. *et al.* From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *arXiv e-prints,* arXiv:1709.03489 (Sept. 2017).

39. Kudo, K. *Constrained quantum annealing of graph coloring* 2018. arXiv: `1806.05782 [quant-ph]`.

40. Bravyi, S., Kliesch, A., Koenig, R. & Tang, E. Hybrid quantum-classical algorithms for approximate graph coloring. *arXiv e-prints,* arXiv:2011.13420. arXiv: `2011.13420 [quant-ph]` (Nov. 2020).

41. Lechner, W., Hauke, P. & Zoller, P. A quantum annealing architecture with all-to-all connectivity from local interactions. *Science advances* **1,** e1500838 (2015).

42. Lechner, W. Quantum approximate optimization with parallelizable gates. *IEEE Transactions on Quantum Engineering* **1,** 1–6 (2020).

43. Henriet, L. Robustness to spontaneous emission of a variational quantum algorithm. *Phys. Rev. A* **101,** 012335. arXiv: `1910.10442 [quant-ph]` (Jan. 2020).

44. Park, J. & Boyd, S. General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming. *arXiv:1703.07870* (2017).

45. Elloumi, S. & Lambert, A. Global solution of non-convex quadratically constrained programs. *Optimization Methods and Software, Taylor & Francis* **34(1),** 98–114 (2019).

46. Billionnet, A., S., E. & A., L. Linear Reformulations of Integer Quadratic Programs. *Modelling, Computation and Optimization in Information Systems and Management Sciences* **14** (2008).

47. Wang, Z., Hadfield, S., Jiang, Z. & Rieffel, E. G. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A* **97.** ISSN: 2469-9934. `http://dx.doi.org/10.1103/PhysRevA.97.022304` (Feb. 2018).

48. *Quantum Learning Machine* `https://atos.net/en/solutions/quantum-learning-machine`.

49. Johansson, J., Nation, P. & Nori, F. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications* **184,** 1234–1240. ISSN: 0010-4655. `http://dx.doi.org/10.1016/j.cpc.2012.11.019` (Apr. 2013).

50. *Belib réseau parisien de bornes de recharges pour véhicules électriques* `www.data.gouv.fr/fr/datasets/belib-reseau-parisien-de-bornes-de-recharges-accelerees-22-kw-ac-dc-pour-vehicules-electriques/`. 2017.

51. Sahni, S. K. Algorithms for Scheduling Independent Tasks. *J. ACM* **23,** 116–127. ISSN: 0004-5411. `https://doi.org/10.1145/321921.321934` (Jan. 1976).

52. De Klerk, E., Pasechnik, D. V. & Warners, J. P. On approximate graph colouring and max-k-cut algorithms based on the θ-function. *Journal of Combinatorial Optimization* **8,** 267–294 (2004).

53. Serret, M. F., Marchand, B. & Ayral, T. Solving optimization problems with Rydberg analog quantum computers Realistic requirements for quantum advantage using noisy simulation and classical benchmarks. *arXiv:2006.11190* (2020).

54. Storn, R. & Price, K. Differential Evolution –A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* **11,** 341–359 (1997).

55. Zhou, L., Wang, S.-T., Choi, S., Pichler, H. & Lukin, M. D. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Phys. Rev. X* **10,** 021067. `https://link.aps.org/doi/10.1103/PhysRevX.10.021067` (2 June 2020).

56. Rocca, P., Oliveri, G. & Massa, A. Differential Evolution as Applied to Electromagnetics. *IEEE Antennas and Propagation Magazine* **53,** 38–49 (2011).

57. Wurtz, J. & Love, P. J. *Bounds on MAXCUT QAOA performance for p>1* 2020. arXiv: `2010.11209 [quant-ph]`.

58. Egger, D. J., Marecek, J. & Woerner, S. *Warm-starting quantum optimization* 2020. arXiv: `2009.10095 [quant-ph]`.

59. Shaydulin, R., Safro, I. & Larson, J. Multistart Methods for Quantum Approximate optimization. *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. `http://dx.doi.org/10.1109/HPEC.2019.8916288` (Sept. 2019).

60. Shaydulin, R. & Alexeev, Y. *Evaluating Quantum Approximate Optimization Algorithm: A Case Study* in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)* (2019), 1–6.

61. Brandao, F. G. S. L., Broughton, M., Farhi, E., Gutmann, S. & Neven, H. *For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances* 2018. arXiv: `1812.04170 [quant-ph]`.

62. Guerreschi, G. G. & Smelyanskiy, M. *Practical optimization for hybrid quantum-classical algorithms* 2017. arXiv: `1701.01450 [quant-ph]`.

63. Nelder, J. A. & Mead, R. A simplex method for function minimization. *Computer Journal* **7,** 308–313 (1965).

64. Gill, P., Murray, W. & Wright, M. *Practical Optimization* ISBN: 9781611975604. `https://books.google.fr/books?id=G-vEDwAAQBAJ` (SIAM, Society for Industrial and Applied Mathematics, 2019).

65. Powell, M. J. D. in *Advances in Optimization and Numerical Analysis* (eds Gomez, S. & Hennart, J.-P.) 51–67 (Springer Netherlands, Dordrecht, 1994). ISBN: 978-94-015-8330-5. `https://doi.org/10.1007/978-94-015-8330-5_4`.