

BBQ-mIS: a parallel quantum algorithm for graph coloring problems

Original

BBQ-mIS: a parallel quantum algorithm for graph coloring problems / Vercellino, Chiara; Vitali, Giacomo; Viviani, Paolo; Giusto, Edoardo; Scionti, Alberto; Scarabosio, Andrea; Terzo, Olivier; Montrucchio, Bartolomeo. - ELETTRONICO. - (2023), pp. 141-147. (2023 IEEE International Conference on Quantum Computing and Engineering (QCE) Bellevue, WA (USA) September 17–22, 2023) [10.1109/QCE57702.2023.10198].

Availability:

This version is available at: 11583/2982390 since: 2023-09-21T18:15:21Z

Publisher:

IEEE COMPUTER SOCIETY

Published

DOI:10.1109/QCE57702.2023.10198

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

BBQ-mIS: a parallel quantum algorithm for graph coloring problems

Chiara Vercellino^{*†‡}, Giacomo Vitali^{*†}, Paolo Viviani^{*}, Edoardo Giusto[†],
Alberto Scionti^{*}, Andrea Scarabosio^{*}, Olivier Terzo^{*}, Bartolomeo Montrucchio[†]

^{*}LINKS Foundation, Torino, Italy

[†]DAUIN, Politecnico di Torino, Torino, Italy

[‡]chiara.vercellino@linksfoundation.com

Abstract—Among the limitations of current quantum machines, the qubits count represents one of the most critical challenges for porting reasonably large computational problems, such as those coming from real-world applications, to the scale of the quantum hardware. In this regard, one possibility is to decompose the problems at hand and exploit parallelism over multiple size-limited quantum resources. To this purpose, we designed a hybrid quantum-classical algorithm, *i.e.*, BBQ-mIS, to solve graph coloring problems on Rydberg atoms quantum machines. The BBQ-mIS algorithm combines the natural representation of *Maximum Independent Set (MIS)* problems onto the machine Hamiltonian with a *Branch&Bound (BB)* approach to identify a proper graph coloring. In the proposed solution, the graph representation emerges from qubit interactions (qubits represent vertexes of the graph), and the coloring is then retrieved by iteratively assigning one color to a maximal set of independent vertexes of the graph, still minimizing the number of colors with the *Branch&Bound* approach. We emulated real quantum hardware onto an IBM Power9-based cluster, with 32 cores/node and 256 GB/node, and exploited an MPI-enhanced library to implement the parallelism for the BBQ-mIS algorithm. Considering this use case, we also identify some technical requirements and challenges for an effective HPC-QC integration. The results show that our problem decomposition is effective in terms of graph coloring solutions quality, and provide a reference for applying this methodology to other quantum technologies or applications.

Index Terms—hybrid quantum-classical optimization, graph coloring, HPC-QC integration, Branch&Bound

I. INTRODUCTION

Recent years have seen a steep rise in interest concerning quantum computers and their integration with High Performance Computing (HPC). For this purpose, it is important to identify both infrastructures to make the classical and quantum machines communicate and applications that can benefit from such an integrated system.

Concerning this last aspect, we propose a hybrid-quantum classical approach to solve graph coloring (GC) problems that emerge from several industrial applications, *e.g.*, optimal deployment of communication networks [1] and analysis of biological networks [2]. The key idea in our algorithm, **BBQ-mIS**, is to combine a classical Branch&Bound algorithm, which by definition is highly parallelizable, with a quantum routine to sample feasible (partial-)coloring solutions.

Among the various quantum technologies, neutral atoms machines [3] are extremely suitable for approaching graph

combinatorial optimization problems, such as GC. They employ Rydberg atoms, *i.e.* neutral atoms, to act as *qubits*, organizing them on a 2D/3D register. Then, the interactions among these atoms, subject to laser pulses [4], generate the machine Hamiltonian H (*e.g.*, Ising)

$$H = \sum_{i=1}^n \frac{\hbar\Omega}{2} \sigma_i^x - \sum_{i=1}^n \frac{\hbar\delta}{2} \sigma_i^z + \sum_{j>i} \frac{C_6}{d_{ij}^6} n_i n_j \quad (1)$$

where, $n_i = (1 + \sigma_i^z)/2$ is the Rydberg state occupancy, $\sigma_i^{x,z}$ are the i -th qubit's Pauli matrices, Ω and δ are respectively the Rabi frequency and the detuning of the controlling laser system, and d_{ij} is the Euclidean distance between qubits i and j in the register. Hence, the way the machine Hamiltonian evolves depends on the laser pulses parameters and on the positions of qubits (neutral atoms), which assume, once measured, one out of two possible quantum states (*i.e.*, the excited Rydberg state $|1\rangle$ or the ground state $|0\rangle$). From this derives the connection with binary optimization variables.

There is also another effect of quantum mechanics that allows the representation of graphs directly on the quantum machine register, *i.e.*, the *blockade* effect. It acts as a threshold on the qubit-to-qubit distance, discriminating the closer ones from the farther ones [5]. This threshold distance is the critical one at which the strength of the qubit-to-qubit interactions balances with the Rabi frequency of the laser pulses [6], and it is called *blockade radius* $r_b = (\frac{C_6}{\hbar\Omega})^{1/6}$. In this way, the interactions between qubits, which can be assimilated to the vertexes of a graph, can induce a Unit-Disk Graph (UDG) [7]: qubit positions in the quantum register are UDG's vertex positions, and edges in the UDG link two vertexes whenever their Euclidean distance is shorter than the blockade radius.

Moreover, the blockade effect prevents qubits, which fall within the blockade radius, from being both in the excited state $|1\rangle$. Thus, retrieving the ground state of H coincides with computing the largest set of non-interacting qubits on the register, *i.e.*, the *Maximum Independent Set (MIS)* of the corresponding UDG.

However, not all the combinatorial optimization problems can be easily reformulated as *MIS* problems on UDGs, or, when this is the case [8], the problem mapping onto the quantum hardware scales badly in the number of physical

qubits to represent the binary variables. Willing to solve the problem of interest and, at the same time, to reduce as much as possible the needed qubits, we designed the **BBQ-mIS** algorithm for GC problems.

More in detail, starting from a UDG representation of the target graph thanks to the methodology presented in [9], the proposed quantum-enhanced heuristic solves the graph coloring problem by iteratively solving *MIS* problems. One could notice that being able to retrieve a UDG from the graph of interest might limit the generality of the approach. However, it is still possible to map one logical qubit into multiple physical qubits to increase the UDG’s connectivity [8] or rely on an approximated UDG (an approximation of the original graph with some missing edges). In fact, our approach does not need to find the optimal *MIS* to solve GC problems.

BBQ-mIS successfully solves graph coloring problems, for all the graph samples of our dataset, with the minimum number of colors. The benchmark solutions are provided by the exact solver *Gurobi* [10].

II. RELATED WORK

Hybrid quantum-classical algorithms keep on emerging in several research areas. They range from the optimization field, *e.g.*, for dynamic portfolio optimization with minimal holding period exploiting *D-Wave 2000Q* processor [11], or, in a sort of complementary direction, the design of optimization methods for parameterized quantum circuits [12], [13], to machine learning, *e.g.*, deep learning time series for ship motion forecasting [14], to material simulation using variational quantum algorithms to determine the ground state of molecular problems by combining classical and quantum (neutral atom machines) hardware [15].

Thus, due to the extensive literature on the topic, we focus on the quantum-classical algorithms to solve the *Graph Coloring (GC)* problem.

The *GC* problem admits a quadratic unconstrained binary optimization (QUBO), so it is possible to solve the corresponding combinatorial optimization problem through quantum annealing. For instance, C. Silva et al. [16] compared the results of solving the QUBO problem of *GC* both with a fully-classical simulated annealing method and using a *D-Wave* quantum machine; according to their study, the quantum approach found more solutions. The paper [17] proposes a similar approach (they still exploit the mapping of the combinatorial optimization problem into chimera graphs and use quantum annealers) but focus more on the effect of an external noise source, which is a hot topic for noisy intermediate-scale quantum devices. In particular, they manage to exploit the noise-generated static disorder to improve performance. Quantum annealing methods for *GC* are also discussed in [18]–[20], but in this case, QUBO formulation is not adopted, constraints and objective function are directly encoded into the problem and driving Hamiltonians.

Quantum circuits for *GC* problems are the topic of [21]. In this work, M. Do et al. extend a previous study on Max-Cut problems [22]: they implement the Quantum Alternating

Operator Ansatz (QAOA) for *GC* on quantum processor architecture. Together with the circuit routing, they propose a qubit initialization approach that allows for shorter makespan compilation. In [23], the authors designed a space-efficient quantum optimization algorithm for the *GC* problem circuit implementation. They approached the problem through a gate-based implementation and showed that gain in the number of logical qubits, which are exponentially reduced in the number of colors, comes to a cost of deeper circuits.

The paper [24] presents a generalization of the Grover algorithm to operate on ternary quantum circuits: the augmented representation capability enhanced by the use of *qudits* instead of *qubits* allows the reduction of the circuit complexity, the simplification of the experimental setup and the enhancement of the algorithm efficiency. Multi-level computational units, specifically *qutrits*, are also present in the work of S. Bravyi et al. [25]. In this case, increasing the state space enables a 3-coloring formulation for *GC* problems.

Finally, the authors in [26] introduce quantum circuits implementing genetic algorithms (GA) to solve NP-hard optimization problems. They apply their methodology to *GC* problems and examine the results obtained with a *Qiskit* simulation environment.

III. METHODOLOGY

In this section, we illustrate two hybrid quantum-classical algorithms. The first one, **Greedy-it-MIS**, is a quantum-enhanced version of a classical heuristic for graph coloring problems; the other one, **BBQ-mIS**, is a novel approach that exploits *MIS* solution to obtain a feasible graph coloring, still minimizing the number of colors used, thanks to a Branch&Bound approach.

Before diving into the descriptions of **Greedy-it-MIS** and **BBQ-mIS** algorithms, we introduce some basic notions of graph theory, thus setting also the notation.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, \mathcal{V} is the set of *vertexes*, $n = |\mathcal{V}|$, \mathcal{E} is the set of undirected *edges*, and A the adjacency matrix of \mathcal{G} . A feasible *coloring* of \mathcal{G} consists of assigning to each vertex in \mathcal{V} a color such that vertexes that share an edge have different colors. A *graph coloring (GC)* problem [27] arises when the feasible coloring of \mathcal{G} is targeted along with the minimization of colors. The number of colors that solve a *GC* problem is called the *chromatic number* of \mathcal{G} , and it is denoted by $\chi(\mathcal{G})$. Another well-known graph combinatorial optimization problem is the *Maximum Independent Set (MIS)* problem. It consists in finding the largest independent set [28] of \mathcal{V} (see def. 1).

Definition 1 (Independent set): Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an *independent set* is a set of vertexes such that no two vertexes share an edge in \mathcal{E} .

A. Greedy-it-MIS for Graph Coloring problems

A first approach to solving *GC* problems, without requiring one optimization variable, *i.e.* a logical qubit, for each possible vertex color assignment [29], and exploiting the straightforward solution of *MIS* problems on neutral atoms, follows from

a simple argument. An *MIS* solution is a feasible color assignment: provided that all vertexes in the *MIS* are independent, they can be colored with the same color. Then, the procedure can iterate over the induced graph $\mathcal{G}' = (\mathcal{V} \setminus MIS, \mathcal{E}')$ that is the original graph \mathcal{G} after removing the vertexes contained in the *MIS* and the corresponding edges. This procedure iterates until all vertexes in \mathcal{V} are associated with a color, as shown in algorithm 1. We named this coloring strategy **Greedy-it-MIS** since we consider the first solution coming from the *MIS solver*, after checking for its independence, to obtain \mathcal{G}' , without evaluating the impact of choosing that particular solution on the overall coloring. In a typical short-sighted way, **Greedy-it-MIS** looks just one step at a time, hence the name *Greedy*. As a result, **Greedy-it-MIS** provides feasible *GC* solutions but does not target the minimization of the colors.

Algorithm 1 Greedy-it-MIS for Graph Coloring problems

Ensure: C is a feasible coloring for graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
 $k \leftarrow 0$ ▷ number of colors used
while $|\mathcal{V}| > 0$ **do**
 if $|\mathcal{E}| > 0$ **then** ▷ \mathcal{V} is not a MIS
 $MIS \leftarrow MIS_solver(\mathcal{G})$
 $\mathcal{G} = (\mathcal{V}, \mathcal{E}) \leftarrow \mathcal{G}' = (\mathcal{V} \setminus MIS, \mathcal{E}')$
 else
 $MIS \leftarrow \mathcal{V}$
 $\mathcal{G} = (\mathcal{V}, \mathcal{E}) \leftarrow \emptyset$
 end if
 $k \leftarrow k + 1$
 $C_k \leftarrow MIS$
end while

B. BBQ-mIS for Graph Coloring problems

Willing to improve the solutions to *GC* problems, one should identify some exploitable directions that **Greedy-it-MIS** does not take into account.

First of all, the probabilistic behavior of quantum systems requires multiple measurements to provide an *MIS* solution. These repeated experiments return a histogram of possible solutions, where the most occurring one is selected as the *MIS* solution by the **Greedy-it-MIS**. However, other solutions in the histogram can yield valuable information, and they come for free by the standard measurement procedure. As a matter of fact, there is no guarantee that the solution to an *MIS* problem is unique, so even when the optimal coloring contains a maximum independent set, it might not be the first one given by the *MIS* solver. Moreover, it happens that optimal colorings do not include *MIS* solutions so, we should also consider smaller independent sets from the histogram of solutions.

Nevertheless, it would be computationally hard to explore by brute-force all possible independent set combinations provided by an *MIS* solver. It is necessary to have a way of cutting out some solutions and evaluating the impact on the targeted *GC* problem of choosing one independent set over another one. An essential building block to deal with this issue is Theorem 1 [30], [31].

Definition 2 (Maximal independent set): A maximal Independent Set (*mIS*) of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an independent set, which is not properly contained in another independent set of \mathcal{G} , i.e., adding one or more vertexes to a maximal independent set would result in losing its independence.

Theorem 1 (Optimal graph coloring): Every graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has an optimal coloring in which (at least) one of the colors is a *maximal independent set*.

Proof: $C = \{C_1, \dots, C_k\}$ vertex sets defining the optimal coloring of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, k colors. Let I be an *mIS* of \mathcal{G} s.t. I contains C_1 then $C' = \{I, C_2 \setminus I, \dots, C_k \setminus I\}$ is still an optimal coloring.

Thanks to this theorem, we can restrict our solutions space investigation to *mIS*s.

At this point, we can set up an effective minimization of colors for a *GC* problem. The proposed solution is a Branch&Bound (BB) approach that branches on *mIS* solutions and considers as bounds the lower bounds on the chromatic number: **BBQ-mIS**.

More precisely, **BBQ-mIS** leverages well-known lower bounds on $\chi(\mathcal{G})$ for the pruning criteria, since the *GC* is a minimization problem, and exploits upper bounds on $\chi(\mathcal{G})$ for the exploration policy of the BB tree. The considered bounds are valid for any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

We denote with $\Delta(\mathcal{G})$ the maximum vertex degree in \mathcal{G} and with d_i the degree of the vertex $i \in \mathcal{V}$, hence the following upper bounds (UB).

- Greedy coloring UB [32]: $\chi(\mathcal{G}) \leq \Delta(\mathcal{G}) + 1 = UB_G$
- Welsh-Powell's UB [33]: $\chi(\mathcal{G}) \leq \max_{i \in \mathcal{V}}(\min(d_i + 1, i)) = UB_{WP}$, provided that $d_1 \geq d_2 \geq \dots \geq d_n$

Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of A , with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, and (n^+, n^0, n^-) the inertia of A , i.e., the number of positive, null and negative eigenvalues respectively, then the following lower bounds (LB) hold.

- Hoffman's LB [34]: $\chi(\mathcal{G}) \geq 1 - (\lambda_1/\lambda_n) = LB_H$
- Elphick-Wocjan's LB [35]: $\chi(\mathcal{G}) \geq 1 + \max(\frac{n^+}{n^-}, \frac{n^-}{n^+}) = LB_{EW}$
- Edwards-Elphick's LB [36]: $\chi(\mathcal{G}) \geq n/(n - \lambda_1) = LB_{EE}$

Putting together these concepts, **BBQ-mIS** represent an outer (with respect to the *MIS* solver) optimization phase, which considers all solutions contained in the histogram coming out from the *MIS* solver and branches over the induced graphs, obtained by removing all the vertexes in one of the parent node's *mIS*s.

At the root of the BB tree, we start with the graph \mathcal{G} to be colored. The BB scheme, as represented in Fig. 1, governs the optimization so that at the end of the procedure we can retrieve as the **BBQ-mIS** solution the best coloring found so far, i.e., the one that requires fewer colors.

At each BB node, the *GC* problem is associated with a state (see Fig. 1a), fully described by a tuple $(\mathcal{H}, C, k, LB, UB)$ where

- \mathcal{H} is the induced subgraph obtained from the one in the parent node's state by removing the *mIS* solution from

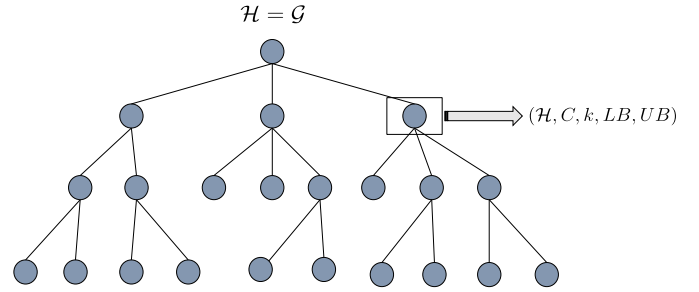
which the branch springs up. \mathcal{H} will be the input graph to the *MIS* solver for the considered BB node off-springs;

- C is a feasible coloring; if the BB node is at depth l ($l = 0$ at the BB root) of the BB tree and $\mathcal{H} = \mathcal{H}(\mathcal{U}, \mathcal{I})$, $m = |\mathcal{U}|$, $C = \{C_1 = mIS_1, C_2 = mIS_2, \dots, C_l = mIS_l, C_{l+1} = u_1, \dots, C_{l+m} = u_m\}$, $u_i \in \mathcal{U}$, i.e., the first l colors are the ones inherited from the parents' *mIS*s and the other m colors are assigned according to the worst-case scenario (one color for each of the remaining vertexes in \mathcal{H});
- k is the value of the objective function in the BB node, i.e., the number of colors in C ;
- LB is a lower bound on $\chi(\mathcal{G})$ by pursuing the exploration of the BB node, i.e., for a node at depth l , $LB = l +$ lower bound on $\chi(\mathcal{H})$; this last lower bound is selected as the tightest among Hoffman, Elphick-Wocjan and Edwards-Elphick lower bounds, that is the $\max(\lfloor LB_H \rfloor, \lfloor LB_{EW} \rfloor, \lfloor LB_{EE} \rfloor)$;
- UB is an upper bound on $\chi(\mathcal{H})$, $UB = \min(\lceil UB_G \rceil, \lceil UB_{WP} \rceil)$

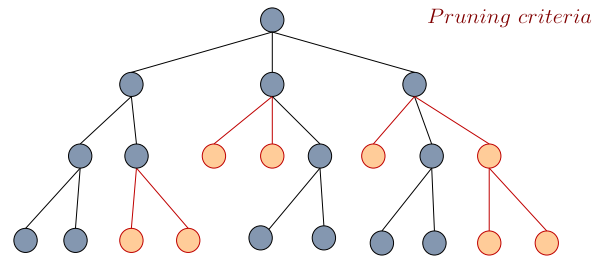
Not all the solutions provided by the *MIS* solver give birth to branches. In fact, the traditional pruning criteria of the BB method are complemented by some others which arise from the problem at hand. The effect of the following pruning criteria (see Fig. 1b) significantly reduce the number of BB nodes that are generated or explored.

- Pruning by non-improving solution: it applies when in the state of a BB node $LB \geq$ best value of the objective function found so far; this is a standard pruning criterion, which prevents us from wasting time and resources by exploring the potential children of a BB node which for sure would not provide a coloring with a smaller number of colors.
- Pruning by unfeasibility: it avoids generating children from the *MIS* solver which are either non-independent sets (they would result in some adjacent vertexes colored with the same color) or non-maximal sets (they would violate Th. 1 on which the overall BB approach relies).
- Pruning by redundancy: it avoids duplicating an exploration in the BB tree; if the same subgraph \mathcal{H} has been generated previously, it will not generate a new BB node. Since the same induced graph can be obtained by removing the same vertexes, but in a different order, we associate with each $\mathcal{H}(\mathcal{U}, \mathcal{I})$ a fingerprint, i.e., $fp(\mathcal{H}) = \sum_{u \in \mathcal{U}} 2^i - 1$, and generate a new BB node only if its fingerprint has not yet been detected.

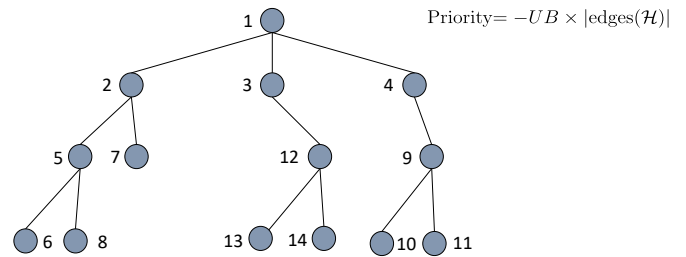
Concerning the BB tree exploration, we first investigated standard policies, such as First-In-First-Out (FIFO), gap-based, and depth-first, then we designed our custom policy that assigns to each BB node a priority score (see Fig.1c). The higher the priority score, the sooner the BB node is explored. To balance between the in-depth and in-width exploration, the priority score is computed as $-UB \times |\mathcal{I}|$. At the top of the BB tree, the priority scores are low. So, in the beginning, the BB exploration goes in-depth, thus providing a best objective



(a) Each node in the Branch&Bound tree is associated with the induced graph \mathcal{H} , the corresponding coloring C , the number of colors k , a lower bound LB on the best coloring achievable by branching further on that node, and an upper bound UB on $\chi(\mathcal{H})$.



(b) The pruning criteria reduce the number of Branch&Bound nodes to explore taking into account non-improving directions, unfeasibility, and equivalences of graphs \mathcal{H} .



(c) A priority score is assigned to each Branch&Bound node to lead the exploration of the Branch&Bound tree: higher priority scores mean that the corresponding node is explored sooner.

Fig. 1. Representation of the Branch&Bound scheme underlying the **BBQ-mIS** algorithm.

function value similar to the one that can be obtained by the *Greedy-it-MIS* algorithm. Then, it starts looking for the most promising directions in the BB tree, the ones with lower connectivity $|\mathcal{I}|$, combined with a tighter bound on the worst-case scenario provided by UB .

Finally, **BBQ-mIS**, as described so far, is designed to end

with its solution once all the BB branches culminate into leaves. However, since the number of *MIS*s scales linearly with the size of the graph n [37] and *MIS*s are computed at each BB node, **BBQ-mIS** could explore a very large number of solutions before ending. Hence, we added a stopping criteria, after which the **BBQ-mIS** returns the best coloring found as its solution. In particular, we set a maximum number of BB nodes that may be explored to 50.

C. Quantum and classical solvers

So far, we have focused mainly on the classical part of our hybrid algorithms, **Greedy-it-MIS** and **BBQ-mIS**. Here, we describe what is the *MIS* solver that brings quantum into the aforementioned methods.

In our experiments, we relied on *Pulser*¹ for the emulation of the quantum system on classical resources. Then, we adopted the Quantum Approximate Optimization Algorithm (QAOA) [4], [38], [39] to optimize the laser pulses' parameters to solve an *MIS* problem. Specifically, the input graph \mathcal{G} is depicted on the array of rearranged Rydberg atoms. The rabi frequency Ω is set according to the desired rydberg radius r_b so that the edges in \mathcal{G} reproduce the proper connectivity. The detuning and the shape of the laser pulses are set according to the specifics of the real hardware *Pasqal's R&D prototype Chadoq2* [40]. The maximum value for rabi frequency and detuning is $12MHz$, and the duration of the pulse sequences is limited to $3\mu s$.

The parameters to optimize, *i.e.*, the variational parameters in QAOA, are the time durations (μs) of the pulses. In this work, we considered just one layer of alternating non-commutative Hamiltonian. The durations' optimization relies on a classical solver (Nelder-Mead simplex algorithm²) that tunes them in order to minimize the energy of the Hamiltonian, so pushing the results towards *MIS*s solutions. Finally, thanks to the Rydberg blockade effect and the tuned laser pulses, we take repeated measurements and obtain a set of viable *MIS*s.

This emulative process can be adopted for graphs with a limited number of vertexes, as it requires solving the differential equation describing the overall quantum system. Practically, it is possible to emulate quantum *MIS* solution on graphs with up to 15 – 17 vertexes within 3 – 20 hours of emulation.

Gurobi classical solver provides a reference value for the *GC* problem solutions. Since the *GC* problems for which it was possible to emulate the quantum dynamics were pretty small, up to 15 vertexes, *Gurobi* solver always return an optimal coloring solution.

D. Remarks about HPC-QC interplay

The application described in this paper represents a good example of a hybrid algorithm for solving combinatorial optimization problems which can leverage both the computational power of modern HPC systems and the characteristics of some types of QPUs, in this case, neutral atoms-based ones. In

fact, the BB is well-suited for parallel computing (our simple implementation already makes use of MPI to launch processes on multiple HPC processes) as each BB node is independent of the others, leading to batches of parallel computation where a set of the BB nodes to be explored is evaluated in parallel. In this context, the HPC produces a large number of quantum-friendly problems that can be solved using a QPU, inducing a few technical challenges related to the interplay between quantum computers and classical computers. Given the early stage of the NISQ platforms, a similar type of usage from typical users (mostly academic and research personnel) is foreseeable [41].

The main challenge is related to parallelism: in the short term, most HPC facilities are expected to have at most one QPU for a given technology (*e.g.*, neutral atoms) available: this represents a first bottleneck for this class of problems where many quantum sub-problems are generated; key parameters, in this case, are related to the relative length of the quantum and classical task and to the latency of moving data around. In this specific case, the quantum task requires 50 *shots*, that are executed by the QPU at a rate that we can assume to be $\sim 5Hz$ [42], [43]. This means that sampling an *MIS* requires around 10s, regardless of the size of the graph; also, *MIS*s can only be addressed one at a time, or a few at a time if considering approaches like mapping multiple graphs on the same QPU register. This granularity is much larger than the time taken by the classical computer to generate problems, creating a bottleneck and wasting all of the advantages of parallelizing the BB tree exploration. On the other hand, the impact of data movement is negligible at this granularity.

The shot rate is then a critical factor to keep into account for an efficient HPC-QC integration. Another related aspect is the scheduling of resources: a low shot rate keeps classical resources idle for a long while waiting for the QPU to complete its task, resulting in a sub-optimal allocation. Possibilities to mitigate this issue range from oversubscribing classical resources when hybrid jobs are involved, to splitting the calculation into multiple steps (*i.e.*, expressing it as a workflow), decoupling the different classical and quantum steps into separate units of scheduling, allowing the batch scheduler to allocate resources to them only when they are actually ready to run. This last approach is particularly suitable also when dealing with cloud-based quantum resources, or with mixed set-ups.

Finally, assuming a different technology (*e.g.*, superconducting) or a dramatically improved shot rate, other resource scheduling challenges should be considered: with only a few QPUs available, we can imagine having a large number of small quantum tasks to be submitted by multiple classical jobs executed by many users. Efficient task queues should be implemented to handle these tasks at a much smaller granularity [44].

IV. RESULTS AND DISCUSSION

To test our algorithms, we created a dataset containing 120 samples of unit-disk graphs with a number of vertexes $n \in$

¹<https://pulser.readthedocs.io>

²<https://docs.scipy.org/>

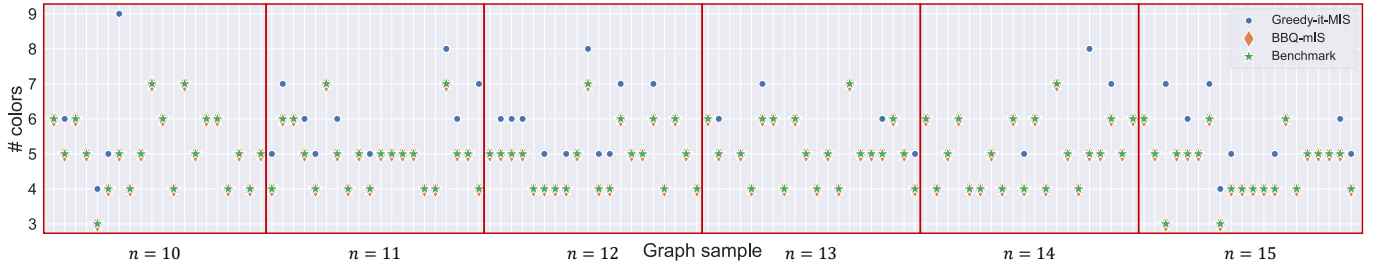


Fig. 2. Comparison of *GC* results: **Greedy-it-MIS** and **BBQ-mIS** algorithms are used to color all graphs in our dataset, *Gurobi* solver provides a benchmark for the coloring solution. The implementation for the *MIS* solver is based on the *Pulser* library.

{10, 11, 12, 13, 14, 15}. Specifically, there are 20 samples for each possible value of n . The unit-disk feasible representation of each graph is obtained with the *GEAN model* presented in [9].

All the experiments were run on an IBM Power9-based cluster, with 32 cores/node and 256 GB/node. In particular, each trial uses 32 physical cores of a node and used up to 16 MPI processes for the parallelization of the Branch&Bound. The overall experiment required ≈ 8000 core hours.

When solving *GC* problems, the *MIS* solver emulates the quantum system, thanks to the *Pulser* library implementation, as discussed in section III-C.

On this dataset, the **BBQ-mIS** algorithm always reaches the coloring solutions with the same number of colors as the optimal *GC* solutions. **Greedy-it-MIS** instead provides worse solutions for 38 samples out of 120; in the worst case, it requires up to 4 colors more (9 colors instead of 5). Fig. 2 summarizes all *GC* results on our dataset.

Another observation concerns the number of nodes used in the Branch&Bound exploration. In the case of our dataset, the maximum number of BB nodes, *i.e.*, 50, is never reached: **BBQ-mIS** terminates into leaves after having explored at most 20 nodes, in the worst-case graph instances. Reasonably, the number of *mIS* coming from small graphs, hence the number of children generating from each branch, is significantly lower than when targeting larger graphs.

When moving to larger graphs and simulations with real quantum hardware, it might be interesting to modify the computation of the feasible coloring C with a less greedy approach than the one which assigns one different color to each of the remaining vertexes in \mathcal{H} . Thus, one could identify a property on the number of vertexes $|\mathcal{U}|$ to establish that when the graph is smaller than a given threshold it is better to solve the subgraph coloring with exact classical methods, possibly even enhanced by HPC, whereas for graph larger than the threshold, the quantum *MIS* solver can be exploited to provide viable *mIS* for partial coloring in a computationally efficient way.

Finally, given the current version of **BBQ-mIS**, and the sampling rate of the quantum machine, we estimated the time needed to solve a *GC*, independently of the graph size. To provide meaningful statistics, we need 50 samples for a fixed pulse setting. This sampling procedure is a minimal quantum

task. To optimize the pulse durations, we allow for evaluating the sampling results 100 times. Once the optimizer finds the best duration values, we retrieve 100 samples to provide the final result of the *MIS* solver. This procedure takes place each time a BB node is explored. Since we limited our optimization to 50 BB nodes in the worst-case scenario, we would need $(50 \times 100 + 100) \times 50 = 255000$ samples to solve a *GC* problem instance. Considering a $5Hz$ sampling rate, the corresponding computational time is ≈ 14 hours. For the specific graph instances of our dataset, this time would be lower because the BB ended by optimality before exploring 50 nodes. In particular, all the corresponding *GC* problems were solved with 8–20 BB nodes, which would correspond to 2–6 hours on the real QPU.

V. CONCLUSION

The work presented so far shows promising results for Graph Coloring problem solutions. Nevertheless, some further directions can be explored to improve **BBQ-mIS**'s performances or provide a broader benchmark.

One possibility is considering the Quantum Adiabatic Algorithm [45] to solve the *Maximum Independent Set* problem instead of the QAOA approach, thus also reducing the use of quantum resources because no pulse optimization phase would be needed.

On the emulation side, it may be interesting to implement the same algorithms exploiting other libraries that emulate the neutral atoms quantum system, such as *Bloqade*³, potentially being able to target even larger graphs. Concerning this aspect, we aim to compare the emulated results with the simulations on real quantum hardware, thus establishing the robustness of our methods in the presence of noise.

Finally, even though these hybrid quantum-classical algorithms target *GC* problems, they can inspire methods with a main focus on saving the required qubits and algorithmic integration with HPC systems, for solving other combinatorial optimization problems.

ACKNOWLEDGMENT

We would like to thank the CINECA consortium for the support and the provisioning of computational resources on the Marconi100 cluster.

³<https://queracomputing.github.io/Bloqade.jl>

REFERENCES

- [1] G. Barillaro, A. Boella, F. Gandino, M. G. Vakili, E. Giusto, G. Mondo, B. Montrucchio, A. Scarabosio, A. Scionti, O. Terzo *et al.*, “Comparison of heuristic approaches to pci planning for quantum computers,” in *2023 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2023, pp. 1–6.
- [2] S. Khor, “Application of graph colouring to biological networks,” *IET systems biology*, vol. 4, no. 3, pp. 185–192, 2010.
- [3] R. Grimm, M. Weidemüller, and Y. B. Ovchinnikov, “Optical dipole traps for neutral atoms,” in *Advances in atomic, molecular, and optical physics*. Elsevier, 2000, vol. 42, pp. 95–170.
- [4] C. Dalyac, L. Henriët, E. Jeandel, W. Lechner, S. Perdrix, M. Porcheron, and M. Veshchezerova, “Qualifying quantum approaches for hard industrial optimization problems. a case study in the field of smart-charging of electric vehicles,” *EPJ Quantum Technology*, vol. 8, no. 1, May 2021. [Online]. Available: <http://dx.doi.org/10.1140/epjqt/s40507-021-00100-3>
- [5] D. Ciampini, O. Morsch, and E. Arimondo, “Ultracold rubidium atoms excited to rydberg levels,” *Journal of Atomic, Molecular, Condensed Matter and Nano Physics*, vol. 2, no. 3, pp. 161–167, 2015.
- [6] C. Picken, R. Legaie, K. McDonnell, and J. Pritchard, “Entanglement of neutral-atom qubits with long ground-rydberg coherence times,” *Quantum Science and Technology*, vol. 4, no. 1, p. 015011, 2018.
- [7] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, no. 1, pp. 165–177, 1990. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0012365X90903580>
- [8] M.-T. Nguyen, J.-G. Liu, J. Wurtz, M. D. Lukin, S.-T. Wang, and H. Pichler, “Quantum optimization with arbitrary connectivity using rydberg atom arrays,” *PRX Quantum*, vol. 4, no. 1, feb 2023. [Online]. Available: <https://doi.org/10.1103/2Fprxquantum.4.010316>
- [9] C. Vercellino, P. Viviani, G. Vitali, A. Scionti, A. Scarabosio, O. Terzo, E. Giusto, and B. Montrucchio, “Neural-powered unit disk graph embedding: qubits connectivity for some qubo problems,” in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 186–196.
- [10] B. Bixby, “The gurobi optimizer,” *Transp. Re-search Part B*, vol. 41, no. 2, pp. 159–178, 2007.
- [11] S. Mugel, M. Abad, M. Bermejo, J. Sánchez, E. Lizaso, and R. Orús, “Hybrid quantum investment optimization with minimal holding period,” *Scientific Reports*, vol. 11, no. 1, p. 19587, 2021.
- [12] K. M. Nakanishi, K. Fujii, and S. Todo, “Sequential minimal optimization for quantum-classical hybrid algorithms,” *Physical Review Research*, vol. 2, no. 4, p. 043158, 2020.
- [13] T. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, B. Grinkemeyer *et al.*, “Multi-qubit entanglement and algorithms on a neutral-atom quantum computer,” *Nature*, vol. 604, no. 7906, pp. 457–462, 2022.
- [14] M.-W. Li, D.-Y. Xu, J. Geng, and W.-C. Hong, “A ship motion forecasting approach based on empirical mode decomposition method hybrid deep learning network and quantum butterfly optimization algorithm,” *Nonlinear dynamics*, vol. 107, no. 3, pp. 2447–2467, 2022.
- [15] R. de Keijzer, O. Tse, and S. Kokkelmans, “Pulse based Variational Quantum Optimal Control for hybrid quantum computing,” *Quantum*, vol. 7, p. 908, Jan. 2023. [Online]. Available: <https://doi.org/10.22331/q-2023-01-26-908>
- [16] C. Silva, A. Aguiar, P. M. Lima, and I. Dutra, “Mapping graph coloring to quantum annealing,” *Quantum Machine Intelligence*, vol. 2, pp. 1–19, 2020.
- [17] A. Wieckowski, S. Deffner, and B. Gardas, “Disorder-assisted graph coloring on quantum annealers,” *Physical Review A*, vol. 100, no. 6, p. 062304, 2019.
- [18] O. Titiloye and A. Crispin, “Parameter tuning patterns for random graph coloring with quantum annealing,” *PLoS one*, vol. 7, no. 11, p. e50060, 2012.
- [19] K. Kudo, “Constrained quantum annealing of graph coloring,” *Physical Review A*, vol. 98, no. 2, p. 022301, 2018.
- [20] O. Titiloye and A. Crispin, “Quantum annealing of the graph coloring problem,” *Discrete Optimization*, vol. 8, no. 2, pp. 376–384, 2011.
- [21] M. Do, Z. Wang, B. O’Gorman, D. Venturelli, E. Rieffel, and J. Frank, “Planning for compilation of a quantum algorithm for graph coloring,” *arXiv preprint arXiv:2002.10917*, 2020.
- [22] B. Mohar and S. Poljak, “Eigenvalues and the max-cut problem,” *Czechoslovak Mathematical Journal*, vol. 40, no. 2, pp. 343–352, 1990.
- [23] Z. Tabi, K. H. El-Safty, Z. Kallus, P. Hága, T. Kozsik, A. Glos, and Z. Zimborás, “Quantum optimization for the graph coloring problem with space-efficient embedding,” in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2020, pp. 56–62.
- [24] Y. Wang and M. Perkowski, “Improved complexity of quantum oracles for ternary grover algorithm for graph coloring,” in *2011 41st IEEE International Symposium on Multiple-Valued Logic*. IEEE, 2011, pp. 294–301.
- [25] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, “Hybrid quantum-classical algorithms for approximate graph coloring,” *Quantum*, vol. 6, p. 678, Mar. 2022. [Online]. Available: <https://doi.org/10.22331/q-2022-03-30-678>
- [26] S. M. Ardelean and M. Udrescu, “Graph coloring using the reduced quantum genetic algorithm,” *PeerJ Computer Science*, vol. 8, p. e836, 2022.
- [27] T. R. Jensen and B. Toft, *Graph coloring problems*. John Wiley & Sons, 2011.
- [28] R. E. Tarjan and A. E. Trojanowski, “Finding a maximum independent set,” *SIAM Journal on Computing*, vol. 6, no. 3, pp. 537–546, 1977.
- [29] F. Glover, G. Kochenberger, and Y. Du, “A tutorial on formulating and using qubo models,” *arXiv preprint arXiv:1811.11538*, 2018.
- [30] A. M. de Lima and R. Carmo, “Exact algorithms for the graph coloring problem,” *Revista de Informática Teórica e Aplicada*, vol. 25, no. 4, p. 57–73, Nov. 2018. [Online]. Available: https://seer.ufrgs.br/index.php/rita/article/view/RITA_Vol25_Nr4_57
- [31] N. Christofides, “An algorithm for the chromatic number of a graph,” *The computer journal*, vol. 14, no. 1, pp. 38–39, 1971.
- [32] J. Mitchem, “On various algorithms for estimating the chromatic number of a graph,” *The Computer Journal*, vol. 19, no. 2, pp. 182–183, 1976.
- [33] D. J. A. Welsh and M. B. Powell, “An upper bound for the chromatic number of a graph and its application to timetabling problems,” *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 01 1967. [Online]. Available: <https://doi.org/10.1093/comjnl/10.1.85>
- [34] A. Hoffman and L. Howes, “On eigenvalues and colorings of graphs. ii,” *Annals of the New York Academy of Sciences*, vol. 175, pp. 238 – 242, 03 2007.
- [35] C. Elphick and P. Wocjan, “An inertial lower bound for the chromatic number of a graph,” *arXiv preprint arXiv:1605.01978*, 2016.
- [36] C. Edwards and C. Elphick, “Lower bounds for the clique and the chromatic numbers of a graph,” *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 51–64, 1983.
- [37] J. M. Byskov, “Enumerating maximal independent sets with applications to graph colouring,” *Operations Research Letters*, vol. 32, no. 6, pp. 547–556, 2004.
- [38] M. F. Serret, B. Marchand, and T. Ayril, “Solving optimization problems with rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks,” *Physical Review A*, vol. 102, no. 5, p. 052617, 2020.
- [39] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv preprint arXiv:1411.4028*, 2014.
- [40] H. Silvério, S. Grijalva, C. Dalyac, L. Leclerc, P. J. Karalekas, N. Shammah, M. Beji, L.-P. Henry, and L. Henriët, “Pulser: An open-source package for the design of pulse sequences in programmable neutral-atom arrays,” *arXiv:2104.15044 [quant-ph]*, Jan. 2022.
- [41] T. S. Humble, A. McCaskey, D. I. Lyakh, M. Gowrishankar, A. Frisch, and T. Monz, “Quantum Computers for High-Performance Computing,” *IEEE Micro*, vol. 41, no. 5, pp. 15–23, Sep. 2021.
- [42] J. Wurtz, A. Bylinskii, B. Braverman, J. Amato-Grill, S. H. Cantu, F. Huber, A. Lukin, F. Liu, P. Weinberg, J. Long, S.-T. Wang, N. Gemelke, and A. Keesling, “Aquila: QuEra’s 256-qubit neutral-atom quantum computer,” Jun. 2023.
- [43] “QAOA and QAA to solve a QUBO problem,” Jul. 2023.
- [44] M. Schulz, M. Ruefenacht, D. Kranzlmüller, and L. B. Schulz, “Accelerating HPC With Quantum Computing: It Is a Software Challenge Too,” *Computing in Science & Engineering*, vol. 24, no. 4, pp. 60–64, Jul. 2022.
- [45] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.