

Sampled-Based Guided Quantum Walk: Non-variational quantum algorithm for combinatorial optimization

Ugo Nzongani^{1,2}, Dylan Laplace Mermoud³, Giuseppe Di Molfetta^{1,4}, and Andrea Simonetto^{2,5}

¹Aix-Marseille Université, CNRS, LIS, 13288 Marseille, France

²UMA, ENSTA, Institut Polytechnique de Paris, Palaiseau, France

³Sorbonne Université, CNRS, LIP6, F-75005, Paris, France

⁴Institut Universitaire de France, Paris, France

⁵CEDRIC, Conservatoire National des Arts et Métiers, 75003 Paris, France

We introduce SamBa-GQW, a novel quantum algorithm for solving binary combinatorial optimization problems of arbitrary degree with no use of any classical optimizer. The algorithm is based on a continuous-time quantum walk on the solution space represented as a graph. The walker explores the solution space to find its way to vertices that minimize the cost function of the optimization problem. The key novelty of our algorithm is an offline classical sampling protocol that gives information about the spectrum of the problem Hamiltonian. Then, the extracted information is used to guide the walker to high quality solutions via a quantum walk with a time-dependent hopping rate. We investigate the performance of SamBa-GQW on several quadratic problems, namely MaxCut, maximum independent set, portfolio optimization, and higher-order polynomial problems such as LABS, MAX- k -SAT and a quartic reformulation of the travelling salesperson problem. We empirically demonstrate that SamBa-GQW finds high quality approximate solutions on problems up to a size of $n = 30$ qubits by only sampling $\text{poly}(n)$ states among 2^n possible decisions. Furthermore, SamBa-GQW compares in par with classically optimized variational approaches, such as the variational guided quantum walk and QAOA (the latter when run on deep circuits). This places SamBa-GQW as a promising heuristic to tackle combinatorial problems beyond the NISQ

regime.

1 Introduction

Combinatorial optimization is a natural benchmark for quantum algorithms. On the one hand, its hard classical nature makes its problems very challenging to solve classically at scale, since the solution space grows exponentially in the number of variables. On the other hand, by its very nature, a quantum computer offers new, compact ways of exploring an exponential space with only a linear number of qubits. As a result, new quantum methods are emerging and may offer an advantage over their classical counterparts [1].

On current noisy devices (NISQ), common approaches for solving combinatorial optimization problems on quantum computers leverage variational quantum algorithms (VQAs) [2–4], which alternate between a classical optimizer and a parametric quantum observable evaluation. This class of algorithms is tailored to current machines, since the depth of the circuit is often a parameter of the algorithm and can be chosen to be small, trading-off accuracy [5].

Among those algorithms, the quantum approximate optimization algorithm (QAOA) [6, 7] is surely the most studied, and it comes with many variants [8–14]. The performance and potential advantage of QAOA is still under debate, but some (albeit limited) empirical evaluations show that it could have the potential of offering scaling advantage over classical methods for some classically intractable problems [15, 16]. However, QAOA, as many (if not all) variational methods, suffers from the presence of barren plateaus [17], which make the classical optimization phase very

Ugo Nzongani: ugo.nzongani@lis-lab.fr

difficult, limiting any potential quantum advantage.

In order to overcome the need for a classical optimization step, we turn to quantum walks. Quantum Walks (QWs) are a coherent transportation model on graphs and a fundamental tool for simulating physics [18–21], and designing quantum algorithms [22–27]. They are a universal model of quantum computation [28, 29], defined in discrete [30] and continuous-time [31]. Furthermore, one can interpret QAOA on n qubits as continuous-time QW on the n -dimensional hypercube where each computational basis state corresponds to a vertex of the hypercube.

Several QWs schemes have been proposed for solving combinatorial optimization problems. In the discrete-time settings, some works propose a quantum version of Metropolis-Hasting algorithm with coined QWs [32, 33]. In the continuous-time setting, QWs were shown to be a good candidate for finding ground states of spin glass problems [34], which is equivalent to finding the optimal solution of an optimization problem. Another example is the quantum walk optimization algorithm (QWOA) [35–37], which combines the traditional QAOA with a mixer based on a continuous-time QW. The QW evolves on a graph that only connects feasible solutions. This latter approach however requires classically enumerating and encoding all feasible solutions, which can be as hard as the original optimization problem.

Other works introduce the concept of multi-stage QWs for tackling optimization problems [38, 39], which consists of alternating several QWs with different piece-wise constant hopping rates. In this context, the guided quantum walk (GQW) scheme [40] is an extension of multistage QWs, and is a mixture between QWs and quantum annealing. The authors show that solving a combinatorial optimization problem is equivalent to finding an optimal hopping rate for the quantum walker to guide it towards high quality solutions. This hopping rate is strictly monotonous, positive, and it depends on the spectrum of the Hamiltonian encoding the cost function. They then turn to a classical optimizer to determine optimal control points of a Bézier curve that models the hopping rate. The numerical results are very encouraging and compare very favorably with respect to quantum annealing and standard

quantum walks; yet, the classical optimization leads to exponential scalings [40] of the number of required iterations. Interestingly, other recent methods also use time-dependent scheduling to solve combinatorial optimization problems by using Planck’s principle rather than adiabatic principle [41], and without any classical optimization [42]. See also [43] for further interesting work on the non-adiabatic regime.

In this paper, we offer the following contributions.

First, we overcome the scaling problems of GQW [40] by introducing an offline classical sampling protocol and present Sampled-Based Guided Quantum Walk (SAMBA-GQW). This algorithm is non-variational: the continuous-time QW is specified after the sampling protocol is run and we do not need any classical optimizer. This avoids the scaling issues and potential barren plateaus.

Second, we present a quantum circuit implementation for SAMBA-GQW whose depth is polynomial with respect to the number of qubits. This step turns the continuous-time QW into a circuit implementable on gate-based quantum computers.

Third, we showcase our algorithm on several combinatorial optimization problems, obtaining consistently high quality approximate solutions, if not the optimal one with high probability. The algorithm compares very well with respect to the original GQW, reducing its execution time by at least one order of magnitude, and in par with QAOA in terms of approximation ratios, when run on deep circuits.

The worst-case time complexity of SAMBA-GQW in the continuous-time regime is inversely proportional to the spectral gap of the problem Hamiltonian, which is characteristic from adiabatic evolutions. As such, the found circuit depth also depends on the spectral gap. However, in our simulations, we observe average-case short evolution times, with respect to the number of qubits, for almost all problems. Furthermore, we observe that one does not have to perform the total evolution for the state vector to be well localized. Thus, for long time evolution, one could only perform a part of the quantum walk evolution, measure prematurely several times to obtain an approximation of the distribution, which is well localized with respect to the initial distri-

bution, and classically recover the best solutions efficiently.

Organization. After a short background section (Section 2), we present our contribution: SAMBA-GQW, in Section 3. We formally introduce the set of combinatorial problems on which we test our algorithm in Section 4, and we showcase our numerical results in Section 5. Lastly, we compare our method with the state of the art: GQW and QAOA in Section 6.

2 Background

We present here the necessary background. Throughout the paper, we will be interested in solving binary optimization problems in the form,

$$x^* \in \arg \min_{x \in \{0,1\}^n} C(x) \quad \text{subject to } x \in \mathcal{F}, \quad (1)$$

where $C : \{0,1\}^n \rightarrow \mathbb{R}$ is the cost, and $\mathcal{F}$ the constraint set. We call x a decision while we label x^* the optimal solution. We also call any $x \in \{0,1\}^n \cap \mathcal{F}$ a feasible decision (i.e., any binary decision that respects the constraints). If x is a feasible decision, then an ϵ -approximate solution ($\epsilon > 0$) is a vector, such that,

$$C(x) - C(x^*) \leq \epsilon.$$

In the following, $C(x)$ will be represented as a polynomial in x , while $\mathcal{F}$ will be either the whole space (in such case, we call the problem unconstrained), or a linear constraint.

2.1 Adiabatic evolution for optimization

The solution of a combinatorial optimization problem as (1) of n binary variables can be expressed as a bitstring $x = \{0,1\}^n$. When the problem is unconstrained, the space of solutions scales exponentially with the size of the problem as there are $N = 2^n$ candidates.

Naturally, one can use n qubits to represent this candidate space as a N -dimensional Hilbert space $\mathcal{H}^{\otimes n}$. One can use then the cost Hamiltonian to apply a bias on each computational basis state as,

$$H_C = \sum_{j=0}^{N-1} C(j) |j\rangle \langle j|. \quad (2)$$

Variational quantum algorithms take inspiration from quantum annealing for solving such

problems. The principle consists of interpolating between two non-commuting Hamiltonians with a specific annealing schedule. Such adiabatic evolutions are of the form [44, 45]:

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_1. \quad (3)$$

The adiabatic theorem of quantum mechanics states that if the system starts in the ground state of H_0 and evolves slowly enough, it will remain in the ground state of $H(t)$ during the evolution. Eventually, it will end up in the ground state of H_1 when $t = T$. Therefore, if the Hamiltonians are chosen such that the ground state of H_0 is easy to prepare and that of H_1 encodes the solution of a combinatorial optimization problem, an adiabatic evolution would lead to the optimal solution [44, 45]. A perfect candidate for the role of the starting Hamiltonian in the unconstrained case is the transverse magnetic field:

$$H_M = - \sum_{j=0}^{n-1} \sigma_j^x, \quad (4)$$

where σ_j^x is the Pauli- X matrix applied on qubit j . The ground state of H_M is the uniform superposition over the computational basis states, which is trivial to prepare with a Hadamard transform on the n input qubits $|0\rangle^{\otimes n}$:

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle. \quad (5)$$

Thus, setting $H_0 = H_M$ and $H_1 = H_C$ in Eq. (3) would induce a continuous-time evolution for solving combinatorial optimization problems. Moreover, since H_M corresponds to the adjacency matrix of an n -dimensional hypercube, this evolution can be interpreted as a continuous-time quantum walk on an hypercube with a vertex-dependent potential field (induced by H_C as it applies a quality-dependent weight on each vertex). Hence, the Hilbert space is spanned by the vertices of the n -dimensional hypercube, i.e. each solution x is associated to vertex $|x\rangle$.

2.2 Guided Quantum Walk

The authors of Ref. [40] revisit the continuous evolution (3) by proposing to use a more general time-dependent hopping rate (later indicated with $\Gamma(t)$) for the annealing schedule, as:

$$H(t) = \Gamma(t) H_M + H_C. \quad (6)$$

To determine an optimal $\Gamma(t)$, they leverage the theory of local amplitude transfer and the energy levels of the Hamiltonians. The intuition is that if the energy levels of $\Gamma(t)H_M$ and H_C are balanced, then the transfer between the two Hamiltonians is the greatest and it is more probable that the evolution favors high quality approximate solutions.

Let Γ be a function of the energy E : $\Gamma = \Gamma(E)$. Then, one can compute the local energy gaps between vertices j and k for H_M and H_C to be $|\Delta_{jk}^M| = 2$ and $\Delta_{jk}^C = |\langle j|H_C|j\rangle - \langle k|H_C|k\rangle|$, respectively. Moreover, the transfer of amplitude from the highest to the lowest local energy state is maximized when the relative strength between H_M and H_C is balanced, i.e. $|\Gamma(E)\Delta_{jk}^M| \approx \Delta_{jk}^C$. By using such balancing condition, one can derive the hopping rate as [40]:

$$\Gamma(E) = \frac{\langle \Delta^C \rangle(E)}{2}, \quad (7)$$

where $\langle \Delta^C \rangle(E)$ is the average of largest energy gaps at energy E .

Now, we can naturally map $\Gamma(E)$ to a function of time: the hopping rate starts at high values and monotonically decreases such that $\Gamma(t = 0 \equiv E_{\max})$ and $\Gamma(t = T \equiv E_{\min})$. This time-dependent hopping rate will sequentially activate different regions of the solution graph (n -dimensional hypercube) and gradually confine the walker to low-energy regions, i.e., optimal solutions. Since Eq. (7) requires knowledge of the spectrum of H_C , its computation is exponential in n . To overcome this issue, the authors of Ref. [40] use an ansatz parametrized with six parameters to approximate the shape of $\Gamma(t)$. However, their approach leads to an exponential scaling of the number of iterations of the variational approach with respect to the problem size, for the classical optimization.

3 Sampled-Based Guided Quantum Walk

We are now ready to propose a one-shot algorithm for solving combinatorial optimization on quantum computers without using any classical optimizer.

The key idea is that one does not need to have the exact spectrum of H_C nor optimize a complex

Algorithm 1: Sampled-Based Guided Quantum Walk (SAMBA-GQW)

Input: Optimization problem of n variables

Output: An approximate solution

- 1 *Classical offline part*
 - 2 Perform the sampling protocol (SAMPLER, Alg. 2) for $q = \text{poly}(n)$ samples
 - 3 Build the annealing schedule $\Gamma(t)$ via BUILDER, Alg. 3
 - 4 *Quantum online part*
 - 5 Run the continuous-time quantum walk evolution of Eq. (14) (or its discrete version Eq. (16))
 - 6 *Measurements*
 - 7 Perform $\text{poly}(n)$ measurements in the computational basis and keep the best solution
-

non-linear curve to obtain satisfactory optimization results. A sufficiently good *estimation* of the spectrum of H_C is enough. This estimation can be obtained classically by sampling only a polynomial number of points, instead of 2^n for the exact spectrum. In fact, we show numerically that n^2 points are sufficient in most cases.

Our algorithm then consists of two parts. The first classical offline part is an efficient sampling protocol for approximating the optimal hopping rate of Eq. (7), while the second quantum online part consists of running the continuous-time evolution of Eq. (6). We also propose a quantum circuit implementation of this evolution for it to be run on a gate-based quantum computer. We present the complete algorithm in Algorithm 1.

3.1 Classical offline part: sampling

Instead of using a variational method, i.e. classical optimization to fine-tune the parameters, we propose to approximate Eq. (7) with a sampling protocol. We compute $\langle \Delta^C \rangle$ for q candidate points $x \in \{0, 1\}^n$. If $q = 2^n$ the computation of Eq. (7) is exact, but exponentially costly; while if $q = \text{poly}(n)$, the computation of Eq. (7) is approximated but computationally advantageous. We will see later that $q = n^2$ will be sufficient¹. At each iteration we proceed as follows, first we randomly select $x \in \mathcal{S} := \{0, 1\}^n$. Then, we compute the energy gap with x and its neighbors states and we store the largest gap with its associated energy. The set of neighbors of x

¹This is not extremely surprising, since we know that smooth curves can be well approximated with poly log inputs [46].

Algorithm 2: SAMPLER

Input: $1 \leq q \leq N = 2^n$ **Output:** Set of energies $\mathcal{E}$

```

1  $\mathcal{S}_0 \leftarrow$  set of candidates with  $|\mathcal{S}_0| \leq 2^n$ 
2  $\mathcal{S}_1 \leftarrow \emptyset$ 
3  $E \leftarrow$  empty dictionary
4  $B(x) \leftarrow$  returns the neighbors of state  $x$ 
5 for  $i \in [1, q]$  do
6   Randomly generate  $x \in \mathcal{S}_0 \setminus \mathcal{S}_1$ 
7    $\Delta \leftarrow \{\Delta_{x,y}^C / |\Delta_{x,y}^M| \mid C(x) > C(y), \forall y \in B(x)\}^a$ 
8    $E[C(x)] \leftarrow \text{mean}(E[C(x)] + \max \Delta)$ 
9   Add  $x$  to  $\mathcal{S}_1$ 
10  $\mathcal{E} \leftarrow$  values of  $E$ 
11 Add 0 to  $\mathcal{E}$ 
12 Remove all duplicates from  $\mathcal{E}$ 
13 Sort  $\mathcal{E}$  in decreasing order
```

^aNotice how we only consider positive energy gaps instead of taking their absolute values, we discuss this choice in Appendix E.

depends on the connectivity of the mixer. In the case of the X -mixer of Eq. (4), it encodes a n -dimensional hypercube, therefore, each state has n neighbors². Lastly, we remove x from $\mathcal{S}$. Once this protocol is performed for the q solutions to sample, we compute the mean of the stored gaps for their corresponding energy value and we obtain an approximation of Eq. (7). Lastly, we interpolate between discrete energy gaps to obtain a continuous and smooth hopping rate as a function of the energy. The formal description of the sampling protocol is presented in Algorithm 2, labelled SAMPLER. We note here the similarity of this protocol to a local neighborhood search in combinatorial optimization.

The complexity of the sampling depends on three elements: the number q of sampled candidates, here $q = \text{poly}(n)$; the maximum number of neighbors per state (which depends on the mixer connectivity), N_{neigh} , and the complexity of classically querying the cost function: C_{cost} . Therefore, the complexity of the sampling protocol is:

$$C_{\text{sampling}} \leq q N_{\text{neigh}} C_{\text{cost}}. \quad (8)$$

Since we are mainly concerned with polynomial problems, we also know that the complexity of querying the cost function of a polynomial problem of degree d is $C_{\text{cost}} = \mathcal{O}(n^d)$.

²In that case, a state $|x\rangle$ is connected to another state $|y\rangle$ if their Hamming distance is 1, i.e. if they only differ by 1 bit.

If we put all this together, the complexity of the sampling is $\text{poly}(n)N_{\text{neigh}}$. For the X -mixer of Eq. (4), we have that $N_{\text{neigh}} = n$, and the whole complexity is still polynomial.

Algorithm 3: BUILDER

Input: List of sampled energy gaps $\mathcal{E}$ sorted in decreasing order**Output:** The hopping schedule $\Gamma(t)$

```

1  $dt\_list \leftarrow [0]$ 
2  $t \leftarrow 0$ 
3 for  $\gamma \in \mathcal{E}$  do
4    $t_\gamma \leftarrow \frac{\pi}{2\sqrt{2}}\gamma^{-1}$  Add  $t_\gamma$  to  $dt\_list$ 
5    $t \leftarrow t + t_\gamma$ 
6   Remove  $\gamma$  from  $\mathcal{E}$ 
7  $\Gamma(t) \leftarrow$  interpolation of  $\mathcal{E}$  where the distance
   between successive sampled points  $\mathcal{E}[i]$  and  $\mathcal{E}[j]$ 
   ( $i < j$ ) is  $dt\_list[i+1]$ 
```

3.2 Hopping rate determination

Having sampled the energy gaps in the Hamiltonians, we are now ready to set the hopping rate. We leverage once more [40]. If we start in a local superposition of states $|j\rangle$ and $|k\rangle$, the probability of measuring the lower energy state $|k\rangle$ in this 2-state system is given by:

$$\mathbb{P}_{jk}(t, \Gamma(E)) = \frac{1}{2} + \frac{\Gamma(E)\delta_{jk}}{\Gamma(E)^2 + \delta_{jk}^2} \sin\left(t\sqrt{\Gamma(E)^2 + \delta_{jk}^2}\right)^2, \quad (9)$$

with the short-hand notation $\delta_{jk} = \Delta_{jk}^C/2$. Moreover, as amplitude transfer from $|j\rangle$ to $|k\rangle$ is maximized when the relative strength between H_M and H_C is balanced, the matching hopping rate is $\Gamma_{jk}^* = \delta_{jk}$. Thus, we can rewrite Eq. (9) as:

$$\mathbb{P}_{jk}(t, \Gamma^*) = \frac{1}{2} + \frac{1}{2} \sin\left(t\Gamma_{jk}^*\sqrt{2}\right)^2. \quad (10)$$

To obtain a high probability of measuring a good low energy solution, we need to find the evolution time which maximizes $\mathbb{P}_{jk}(t, \Gamma^*)$. This corresponds to solving $\mathbb{P}_{jk}(t, \Gamma^*) = 1$:

$$t_{jk}^* = \frac{\pi}{2\sqrt{2}} \left(\Gamma_{jk}^*\right)^{-1}, \quad (11)$$

in the following, we refer to this time as the *transfer time*.

In order to maximize the probability of measuring *all* the lowest-energy states, i.e. optimal

solutions, one should run the quantum walk evolution of Eq. (6) for a total duration:

$$T^* = \frac{\pi}{2\sqrt{2}} \sum_{(jk) \in E_M} (\Gamma_{jk}^*)^{-1}, \quad (12)$$

with E_M the set of edges of the mixer underlying graph. In this way, the walker has time to hop through all the energy gaps. Each term of the sum in Eq. (12) is upper bounded by Δ^{-1} , where Δ the spectral gap of H_C , i.e. the modulus difference between its ground and first excited states. Therefore, the scaling of the exact time evolution is $T^* = \mathcal{O}(\Delta^{-1})$. In addition, in the case of the n -dimensional hypercube, $|E_M| = n2^{n-1}$, so that $T^* = \mathcal{O}(n2^{n-1}\Delta^{-1})$.

In practice, one does not have access to the perfect hopping rate Γ^* , but rather to that of Eq. (7), which corresponds to the average of the largest energy gaps. Additionally, we approximate Γ by sampling the cost function. Therefore, we employ the approximation of T^* as,

$$T = \frac{\pi}{2\sqrt{2}} \sum_{(jk) \in E_S} \Gamma_{jk}^{-1}, \quad (13)$$

with E_S being the set of sampled edges. As we approximate Γ using a number of samples that is polynomial in n , we obtain $T = \mathcal{O}(\Delta^{-1}\text{poly}(n))$.

Finally, as seen in Eq. (11), the time input of Γ depends on the energy. With the rescaling $t_{ij} = \Gamma_{jk}^{-1}$, we can write the evolution as:

$$H(t_{jk}) = \Gamma \left(\frac{\pi}{2\sqrt{2}} t_{jk} \right) H_M + H_C. \quad (14)$$

We illustrate the exact and approximated sampling, and resulting interpolation of the hopping rate for $n = 11$ qubits on Fig. 1, and the complete algorithm in Algorithm 3, labelled BUILDER. We note that the approximate and the exact $\Gamma(E)$ are close as a function of energy, but not necessarily as a function of time, since $\Gamma(t)$ is rescaled with maximal time dependent on the sum of all energies (Eq. 13). The exact $\Gamma(t)$ decays therefore slower than the approximate one, which is a side advantage for us. Moreover, we present the hopping rate as a function of energy, instead of time, in Appendix A.

Once the hopping schedule is computed, then we can run the quantum walk and perform measurements. In the continuous-time regime, this implies running (14) for a time T . In a gate-based quantum computer, this would mean discretize in time the evolution and implement a circuit.

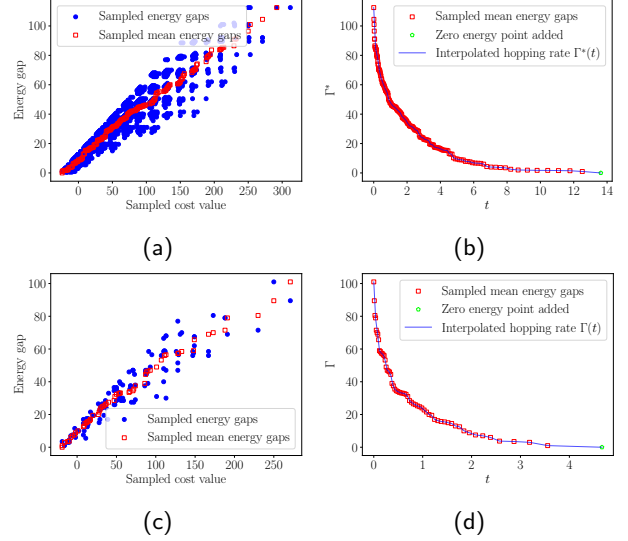


Figure 1: Exact (top) and approximated (bottom) construction of the hopping rate Γ for $n = 11$ qubits (for UD-MIS instance). (a) Exact sampling with $q = 2^n$ samples. (b) Linear interpolation of the hopping rate Γ . The optimal evolution time is given by Eq. (12). The last interpolated point is the zero energy added in Alg. 2 to ensure $\Gamma(t = T) = 0$. (c) Approximated sampling with $q = n^2$ samples. (d) Linear interpolation of the hopping rate Γ . The approximated evolution time is given by Eq. (13).

3.3 Circuit implementation

Having described our main continuous-time quantum walk, we now turn to the question of how to implement it on a gate-based quantum computer.

Because our Hamiltonian is not a simple convex combination of simpler Hamiltonians due to the hopping rate, the discretization of the dynamics via trotterization is not trivial. The key to our approach (to maintain the circuit depth manageable and the accuracy reasonable) is to have two nested discretizations of the time interval. First we divide the time interval into periods τ_ℓ 's that represents the time in which the approximate hopping rate is linear. Then, we subdivide these periods into smaller time intervals p_ℓ 's, on which we apply the trotterization. We can see a depiction of our method in Figure 2. This way of proceeding yields better accuracy than a global trotterization on the Hamiltonian (with a constant sampling period, often employed in QAOA) and it is consistent with the interpolation of the hopping rate.

With this in mind, the evolution we aim for is described by the time-dependent Schrödinger

equation:

$$i\hbar\partial_t |\psi(t)\rangle = H(t) |\psi(t)\rangle, \quad (15)$$

with $|\psi(t)\rangle = U(t) |\psi(0)\rangle$ and we work in units in which $\hbar = 1$.

Since the Hamiltonian is time-dependent, the corresponding propagator $U(t)$ is the time-ordered exponential

$$U(t) = \mathcal{T} \exp \left(-i \int_0^t H(s) ds \right).$$

Here, $\mathcal{T}$ denotes the *time-ordering operator*, which acts on a product of time-dependent operators by reordering them chronologically, with later times place to the left. Formally, for two time-dependent operators A and B ,

$$\mathcal{T}[A(t)B(s)] = \begin{cases} A(t)B(s) & \text{if } t \geq s, \\ B(s)A(t) & \text{if } s > t. \end{cases}$$

The time-ordered exponential is then understood as the limit of the ordered product

$$\begin{aligned} \mathcal{T} \exp \left(-i \int_0^t H(s) ds \right) \\ = \lim_{p \rightarrow \infty} \prod_{\ell=p}^1 \exp \left(-i \frac{t}{p} H(t_\ell) \right), \end{aligned}$$

where $t_\ell = \ell t/p$ and the product is ordered with $\ell = p$ on the left, reflecting the fact that later-time factors act last on the state. When $H(t)$ and $H(s)$ commute for all t and s , the time-ordering is useless and the expression reduces to the ordinary exponential $\exp \left(-i \int_0^t H(s) ds \right)$.

To approximate this on a quantum circuit, we follow a two-level discretization strategy. Let $\mathcal{E}$ be the set of all energy levels corresponding to the values that the hopping rate can take, ordered in a decreasing order. From this set, we define $\mathcal{D}$ as the set of all durations τ_ℓ required for the optimal evolution when the hopping rate takes the value $e_\ell \in \mathcal{E}$. Then, the elements of $\mathcal{D}$ dissect the interval $[0, t]$ with $t = \sum_{\tau \in \mathcal{D}} \tau$ into subintervals

$$T_\ell = \left[\sum_{j=0}^{\ell-1} \tau_j, \tau_\ell + \sum_{j=0}^{\ell-1} \tau_j \right],$$

where $\tau_0 = 0$. The two sets $\mathcal{E}$ and $\mathcal{D}$ have cardinality q , the number of samples³. At the coarse

³In reality, as duplicates of sampled energies are removed, these sets can contain fewer than q elements. Thus, we have $|\mathcal{E}| = |\mathcal{T}| \leq q$.

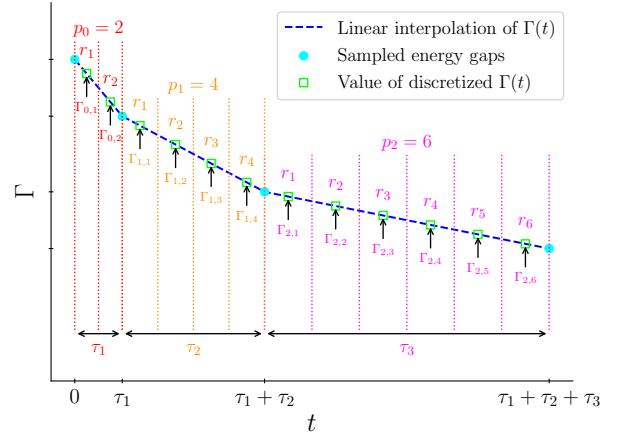


Figure 2: Discretization of the hopping rate Γ with $p = (p_0 = 2, p_1 = 4, p_2 = 6)$. Each interval T_ℓ is cut into p_ℓ subintervals indexed r_i with $i \in \{1, \dots, p_\ell\}$. The value of the discretized hopping rate $\Gamma_{\ell,r}$ corresponds to the average in each subinterval r . The higher the values in p , the more accurate the approximation of Γ . Each subinterval r_i corresponds to the application of a single layer of cost and mixer Hamiltonians, thus the number of layers in the circuit is $\bar{p} = \sum_\ell p_\ell$.

level, the intervals T_ℓ are chosen so that Γ is well-approximated by a linear interpolation on each T_ℓ . At the fine level, each T_ℓ is subdivided into p_ℓ slices of duration τ_ℓ/p_ℓ , on each of which Γ is approximately constant. On such a slice, the Hamiltonian is therefore approximately time-independent, and the first-order Trotter-Suzuki formula [47, 48] yields a controlled approximation.

For $r \in [p_\ell]$, we denote by $U^{(p,r)}(T_\ell)$ the unitary approximating

$$U^{(p,r)}(T_\ell) \simeq U \left(\left[\tau_{<\ell} + r \frac{\tau_\ell}{p_\ell}, \tau_{<\ell} + (r+1) \frac{\tau_\ell}{p_\ell} \right] \right),$$

with $\tau_{<\ell} := \sum_{j<\ell} \tau_j$. On the subinterval

$$\left[\tau_{<\ell} + r \frac{\tau_\ell}{p_\ell}, \tau_{<\ell} + (r+1) \frac{\tau_\ell}{p_\ell} \right],$$

the average value of the hopping rate is given by

$$\Gamma_{\ell,r} = \Gamma_\ell - \left(r + \frac{1}{2} \right) \left(\frac{\Gamma_\ell - \Gamma_{\ell+1}}{p_\ell} \right).$$

Therefore, we choose the unitary $U^{(p,r)}(T_\ell)$ to be

$$U^{(p,r)}(T_\ell) = e^{-i \frac{\tau_\ell}{p_\ell} \Gamma_{\ell,r} H_M} e^{-i \frac{\tau_\ell}{p_\ell} H_C}.$$

In this way, the larger all the p_ℓ are, the more the evolution follows the linear interpolation of the

values of the hopping rate we sampled earlier. We display the discretization process of the hopping rate in Fig. 2. The whole circuit can be expressed in the following form

$$U(t) \simeq \prod_{\ell=1}^q \left(\prod_{r=1}^{p_\ell} U^{(p,r)}(T_\ell) \right). \quad (16)$$

To decompose this circuit in terms of primitive gates, we split each

$$U^{(p,r)}(T_\ell) = U_M^{(p,r)}(T_\ell) U_C^{(p)}(T_\ell)$$

with

$$\begin{cases} U_M^{(p,r)}(T_\ell) = \exp\left(-i \frac{\tau_\ell}{p_\ell} \Gamma_{\ell,r} H_M\right), \\ U_C^{(p)}(T_\ell) = \exp\left(-i \frac{\tau_\ell}{p_\ell} H_C\right). \end{cases}$$

The quantum circuit implementation of $U_C^{(p)}(T_\ell)$ is quite standard, and consists of CNOT and R_Z for QUBO [49] and HUBO [50] problems. As for the mixer $U_M^{(p,r)}(T_\ell)$, its implementation is straightforward, even if it depends on the hopping rate and on the connectivity that we select. For a hypercube, it can be computed as:

$$U_M^{(p,r)}(T_\ell) = \bigotimes_{b=0}^{n-1} R_X^{(b)} \left(-2 \frac{\tau_\ell}{p_\ell} \Gamma_{\ell,r} \right). \quad (17)$$

Note that for maximization problems, the minus sign in the arguments of the rotation gate must be discarded. We depict an example of the quantum circuit implementation in Figure 3.

One may notice that the unitary of Eq. (16) induces a QAOA-like evolution. However, unlike QAOA-like evolutions, we do not have to use a classical optimizer to fine-tune the parameters as we guide the walker to the optimal solutions [40].

The depth of the circuit is:

$$\begin{aligned} d(n, p) &= d_{SP}(n) + \sum_{l=1}^q p_l (d_M(n) + d_C(n)) \\ &= d_{SP}(n) + \bar{p} (d_M(n) + d_C(n)), \end{aligned} \quad (18)$$

where $\bar{p} = \sum_{\ell=1}^q p_\ell$, $d_{SP}(n)$ and $d_M(n)$ are the depth of the implementation of the state preparation and the mixer unitary, and $d_C(n)$ is the depth of a single layer application of the cost Hamiltonian, which can be reduced by adding ancillary qubits [51]. In the case of X -mixer, we have $d_{SP}(n) = d_M(n) = 1$.

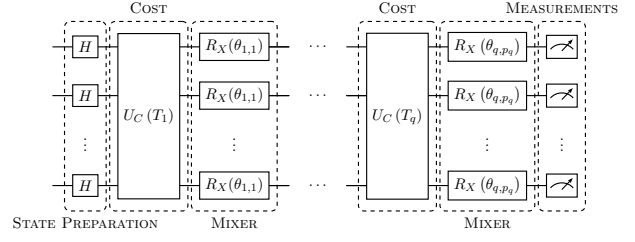


Figure 3: Quantum circuit implementation of the algorithm with X -mixer. The circuit is QAOA-like since it contains layers of cost U_C and mixer U_M unitaries. The angle of the mixer unitaries depends both on l and r , while the cost unitaries depend only on l . The value of the angle $\theta_{l,r}$ is given in Eq. (17). The circuit contains $\bar{p} = \sum_l p_l$ cost and mixer Hamiltonians layers.

In practice, we take all the p_ℓ equal, i.e., $p_\ell = p_\star$ for all $\ell \in [q]$ (see Appendix B). Then, we have that $\bar{p} = qp_\star$, hence

$$d(n, p) = d_{SP}(n) + qp_\star (d_M(n) + d_C(n)). \quad (19)$$

Therefore, the depth of the circuit depends on the number of samples q , the number of steps in the discretization, and the depth of the cost Hamiltonian, that is problem-specific.

We present the general algorithm to construct the quantum circuit in Alg. 4. In Section 4, we give the depth of the circuit according to the problem under study.

Algorithm 4: Quantum circuit

Input: List of sampled energy gaps $\mathcal{E}$ sorted in decreasing order, list p of layer numbers

Output: A quantum circuit implementation of the continuous-time quantum walk

- 1 Add state preparation of initial state (depends on H_M)
 - 2 **for** $l \in \{0, \dots, |\mathcal{E}| - 1\}$ **do**
 - 3 **for** $r \in \{1, \dots, p_\ell\}$ **do**
 - 4 $\Gamma_\ell \leftarrow \mathcal{E}_\ell$
 - 5 $\tau_\ell \leftarrow \Gamma_\ell^{-1} \cdot \frac{\pi}{2\sqrt{2}}$
 - 6 $\Gamma_{\ell,r} \leftarrow \Gamma_\ell - \left(r + \frac{1}{2}\right) \cdot \frac{\Gamma_\ell - \Gamma_{\ell+1}}{p_\ell}$
 - 7 Add $\exp(-i\Delta_t^{(\ell)} H_C)$ for $\Delta_t^{(\ell)} = \tau_\ell / p_\ell$
 - 8 Add $\exp(-i\Delta_t^{(\ell,r)} H_M)$ for $\Delta_t^{(\ell,r)} = \Gamma_{\ell,r} \cdot \tau_\ell / p_\ell$
 - 9 Add measurements in the computational basis
-

4 Set of problems

In this section, we give a brief description of the combinatorial optimization problems on which we

test our algorithm. While some problems are well studied, some technicalities for the less studied ones need to be discussed before we present the simulations.

4.1 Quadratic problems

The first set of problems we study are quadratic unconstrained binary optimization problems, or QUBOs. These are combinatorial optimization problems for which the general problem form (1) has quadratic cost,

$$C(x) = x^\top Qx = \sum_i Q_{ii}x_i + \sum_{i<j} Q_{ij}x_ix_j + c,$$

for a $Q \in \mathbb{R}^{n \times n}$, and $\mathcal{F}$ is the whole space. For one of the selected problems, we will also add a linear constraint.

It is well-known that finding the optimal solution of a QUBO is equivalent to finding the ground state of its Ising formulation, which can be in turn well implemented into a quantum computer.

As instances of QUBO problems we study the following ones.

4.1.1 MaxCut

The weighted Maximum Cut (MaxCut) problem consists of partitioning the vertices of a graph into two disjoint sets so that the number (or weight) of edges between the sets is maximized. This implies defining a graph among the n variables and selecting the cost function,

$$C(x) = - \sum_{i<j} w_{ij}(x_i + x_j - 2x_ix_j), \quad (20)$$

where w_{ij} is the weight of edge (i, j) connecting the variables i and j .

4.1.2 Maximum Independent Set

For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the Maximum Independent Set (MIS) problem consists of finding the independent subset of vertices that maximizes the weight. A subset $S \subseteq \mathcal{V}$ is said to be independent if no two vertices of S are adjacent. Each vertex i is associated to a binary variable $x_i \in \{0, 1\}$, thus we have $n = |\mathcal{V}|$. Moreover, we set $x_i = 1$ if $i \in S$. Therefore, cost function of MIS is:

$$C(x) = - \sum_{i \in \mathcal{V}} w_i x_i + \lambda \sum_{(i,j) \in \mathcal{E}} x_i x_j, \quad (21)$$

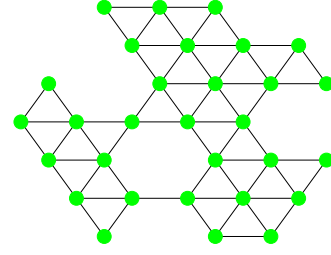


Figure 4: Unit-Disk (UD) graph instance of $n = 30$ vertices.

where w_i is the weight of vertex i and $\lambda > 0$ is the penalty coefficient for including adjacent vertices in S . To ensure that the final set is independent, we define the penalty coefficient such that:

$$\lambda = \max_{(i,j) \in \mathcal{E}} (w_i + w_j) + 1. \quad (22)$$

The study of the MIS problem on quantum computers is of growing interest and particularly on Unit-Disk (UD) graphs [52], as they are native to neutral-atom technologies that exploit the Rydberg blockade mechanism [53–56]. In this context, the problem is often referred to as UD-MIS, moreover, its hardness was first assessed [57] and then shown to strongly depend on the graph density, tree-width and thickness [58]. Interestingly, recent work introduces an imaginary-time evolution scheme for solving UD-MIS [59].

4.1.3 Portfolio optimization: constraints

Portfolio optimization consists of selecting at most k assets among n to obtain the optimal portfolio. Each asset has an expected return and one aims at maximizing profit while minimizing financial risk. Each asset i is associated to a binary variable $x_i \in \{0, 1\}$ and $x_i = 1$ if asset i is in the portfolio. The limitation in the number of assets corresponds to bounding the Hamming weight of the solutions with a fixed value $k \in [0, n]$, i.e. choosing a constraint set as,

$$\mathcal{F} = \left\{ x \in \{0, 1\}^n \mid \sum_{i=1}^n x_i = k \right\}.$$

The cost function reads:

$$C(x) = \lambda \sum_{i<j} \sigma_{ij} x_i x_j - \sum_i \mu_i x_i, \quad (23)$$

with $\sigma \in \mathbb{R}^{n \times n}$ the covariance matrix between assets, $\mu \in \mathbb{R}^n$ is the expected return of the assets and $\lambda > 0$ the risk appetite.

One way of encoding constraints is to use specific mixer Hamiltonian that can allow transitions only between feasible solutions. The Hamming weight constraint can be encoded with XY -mixers [8, 60], which only induce transitions between states with same Hamming weight. Moreover, the connectivity of XY -mixers, i.e. connectivity between states of same Hamming weight, is given by an adjacency matrix A_{XY} . Therefore, such mixers Hamiltonian read:

$$H_{XY} = -\frac{1}{2} \sum_{(i,j) \in A_{XY}} \sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y. \quad (24)$$

Moreover, an optimal quantum circuit implementation of such permutation mixers is proposed in Ref. [4].

Given two qubits a and b such that $|a\rangle \otimes |b\rangle \in \{|i\rangle, |j\rangle, |k\rangle, |l\rangle\}$, XY -mixers act locally as:

$$\begin{aligned} H_{XY}^{(i,j,k,l)} &= -\frac{1}{2} ((\sigma_a^x \otimes \sigma_b^x) + (\sigma_a^y \otimes \sigma_b^y)) \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned} \quad (25)$$

we see that it acts as a negative Pauli- X gate on basis states $|j\rangle$ and $|k\rangle$. In this context, the local cost Hamiltonian is:

$$H_C^{(i,j,k,l)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \langle j| H_C |j\rangle & 0 & 0 \\ 0 & 0 & \langle k| H_C |k\rangle & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (26)$$

Since transfer between higher energy state j to lower energy state k is maximized when $|\Gamma^* \Delta_{jk}^M| \approx \Delta_{jk}^C$, the general form of the hopping rate reads:

$$\Gamma^*(E_{jk}) = \frac{\Delta_{jk}^C}{|\Delta_{jk}^M|}. \quad (27)$$

Looking at Eq. (25), we can see that the local energy gap of Hamiltonian $H_{XY}^{(j,k)}$ is $|\Delta_{jk}^{XY}| = 2$, coincidentally the same as for the X -mixer. Therefore, the hopping rate for XY -mixer is the same as the one in Eq. (7).

Furthermore, the initial state cannot be the uniform superposition over the computational basis states as some of them correspond to infeasible solutions. The correct initial state is the uniform superposition over the basis states of Hamming

weight smaller or equal to the bound h . Previously in the sampling protocol, after selecting a random state, we computed its energy gap with its n neighbors as the underlying graph of the X -mixer is the n -dimensional hypercube. In the case of XY -mixers, the number of neighbors depends on the connectivity of the mixer, which is defined by A_{XY} . Thus, the complexity of the sampling protocol depends on the connectivity of the mixer.

In our experiments we use XY -mixers with ring connectivity. The underlying XY -mixer graph is a disconnected graph that contains $n+1$ subgraphs, each composed of $\binom{n}{k}$ vertices of same Hamming weight k . Ring connectivity implies that two states are connected if they only differ by a single permutation (swap) of two adjacent⁴ bits with different values. Therefore, the maximum number of neighbors is n , and it is achieved for bitstrings in which every pair of adjacent bits differs. Thus, the asymptotic complexity of the sampling protocol for XY -mixers with ring connectivity is the same as for X -mixer, i.e.

$$\mathcal{C}_{\text{sampling}}^{XY_{\text{ring}}} = \mathcal{C}_{\text{sampling}}^X = \mathcal{O}(\text{poly}(n)).$$

As an example, we show the underlying graph of XY -mixer with ring connectivity on $n = 5$ qubits in Fig. 5. The set of neighbors of bitstring x can be formally expressed as $B(x) = \{y \in \{0,1\}^n \mid y_i \neq y_{(i+1) \bmod n}, y = \text{swap}_{i,(i+1) \bmod n}(x)\}$, where $\text{swap}_{i,j}(x)$ is the bitstring obtained by exchanging the values of bits i and j in x . Additionally, the initial state $|\psi_k\rangle$ is the k -Dicke state [61], i.e. the uniform superposition over basis states with Hamming weight equal to k :

$$|\psi_k\rangle = \frac{1}{\sqrt{|F_k|}} \sum_{x \in F_k} |x\rangle, \quad (28)$$

where $F_k = \{x \in \{0,1\}^n \mid \sum_{i=1}^n x_i = k\}$ is the set of bitstring satisfying the k -Hamming weight constraint, i.e. the set of feasible solutions.

4.2 Polynomial problems

Higher-order unconstrained binary optimization (HUBO) problems⁵ are a generalization of

⁴Note that we have boundary conditions such that the last bit is adjacent to the first one since it is a ring connectivity.

⁵Also referred to as polynomial unconstrained binary optimization (PUBO) or higher-order unconstrained bi-

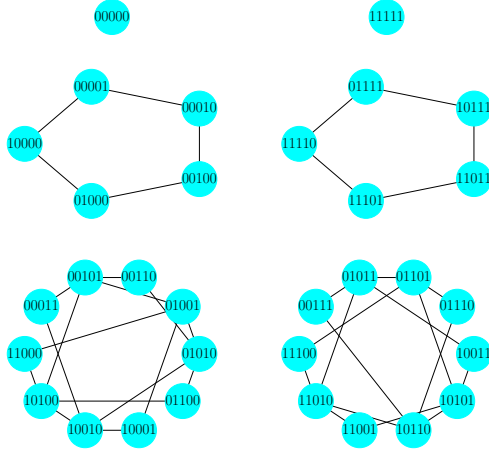


Figure 5: Underlying XY-mixer graph with ring connectivity on $n = 5$ qubits. The graph is disconnected and contains $n + 1 = 6$ subgraphs of different Hamming weights $k \in \{0, 1, 2, 3, 4, 5\}$. Subgraph of Hamming weight k contains $\binom{n}{k}$ vertices. Two vertices x and y are connected if they only differ by a swap of two adjacent different bits, i.e. if $y = \text{swap}_{i,(i+1) \bmod n}(x)$ and $x_i \neq x_{(i+1) \bmod n}$.

QUBO as they describe arbitrary polynomial problems. They are special cases of Problem (1), when the cost is selected as a polynomial,

$$C(x) = \sum_{m \in \pi} \left(\alpha_m \prod_{i \in m} x_i \right), \quad (29)$$

where π is the polynomial set of monomials of at most n variables and $\alpha_m \in \mathbb{R}$ is the coefficient of monomial m . Here, the set $\mathcal{F}$ is selected to be the whole space.

4.2.1 LABS

The low autocorrelation binary sequences (LABS) is a quartic combinatorial problem [62]. This problem is known to be intractable classically as its difficulty increases exponentially with the size n of the sequence. Moreover, it has been shown that QAOA has the potential to outperform classical approaches to solve this problem [15]. Given a sequence $s = (s_1, \dots, s_n)$ with $s_i = \pm 1$, one aims at minimizing its energy:

$$E(s) = \sum_{k=1}^{n-1} \left(\sum_{i=1}^{n-k} s_i s_{i+k} \right)^2. \quad (30)$$

nary optimization (HUBO) problems.

Equivalently, the problem can be reduced to maximizing the merit factor $F(s) = n^2/2E(s)$. A sequence s of Ising variables can be mapped to Boolean variables $x = (x_1, \dots, x_n)$ with the relation $s_i = 1 - 2x_i$. Thus, the cost function reads:

$$C(x) = \sum_{k=1}^{n-1} \left(\sum_{i=1}^{n-k} 4x_i x_{i+k} - 2(x_i + x_{i+k}) + 1 \right)^2. \quad (31)$$

4.2.2 MAX- k -SAT

The boolean satisfiability problem (SAT) is a famous problem in theoretical computer science as it was the first problem proven to be NP-complete [63]. Several works try to solve this problem with variants of QAOA [8, 16, 64, 65]. Given m clauses φ_i of k literals each, one has to decide if there exists an assignment of n variables that satisfies all the clauses. The combinatorial problem of finding such an assignment is called MAX- k -SAT. Formally, a boolean formula in conjunctive normal form of m clauses reads:

$$\phi = \varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_m. \quad (32)$$

Each clause φ_j is a disjunction of k literals l_{jk} :

$$\varphi_j = (l_{j1} \vee l_{j2} \vee \dots \vee l_{jk}), \quad (33)$$

with $l_{ji} \in \{x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, x_n, \bar{x}_n\}$ where $\bar{x}_j = 1 - x_j$. The MAX- k -SAT consists of finding an assignment $x = x_1 x_2 \dots x_n$ that maximizes the number of satisfied clauses. A clause is said to be satisfied if it contains at least one literal equal to 1. This criterion can be expressed as:

$$S_j(x) = 1 - \prod_{i=1}^k (1 - l_{ji}). \quad (34)$$

Therefore, a clause φ_j is satisfied with assignment x if $S_j(x) = 1$, and unsatisfied if $S_j(x) = 0$. One aims at maximizing the number of satisfied clause, thus, the cost function to minimize is:

$$C(x) = - \sum_{j=1}^m S_j(x). \quad (35)$$

This problem is a polynomial of degree k .

4.2.3 Travelling salesperson problem

The travelling salesperson problem (TSP) is an NP-complete decision problem with many applications [66]. Given a list of n cities with specific distance between each of them, is there a route with total distance k that visits all of cities exactly once and returns to the starting point. The related optimization problem consists of finding the shortest route that visits every city exactly once. Formally, the problem is represented with a graph whose vertices and edges respectively correspond to the cities and path between them. The distance between each pair of cities is given by the weight of their connecting edge. Its original QUBO encoding uses n^2 qubits [49]. However, one can obtain a more efficient encoding with $n\lceil\log_2(n)\rceil$ qubits at the cost of expressing TSP as a HUBO [67]. In this higher-order encoding, each city is encoded in binary with $\lceil\log_2(n)\rceil$ bits. Thus, one needs n of these sequences to list all cities. Therefore, each assignment x has two labels, x_i is the i -th bitstring sequence and $x_{i,j}$ is the j -th bit of sequence x_i . We provide an example of this encoding in Fig. 6. The cost function of the HUBO encoding of TSP reads [67]:

$$C(x) = \lambda \sum_{i=0}^{n-1} C_1(x_i) + \gamma \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} C_2(x_i, x_j) + \mu \sum_{i,j=0, i \neq j}^{n-1} w_{ij} \sum_{k=0}^{n-1} C_2(x_k, i) C_2(x_{k+1}, j), \quad (36)$$

where w_{ij} is the weight of edge (i, j) and $\lambda, \gamma > \mu \max_{(i,j)} w_{ij}$. The subcost C_1 puts a penalty if a sequence encodes an invalid city, i.e. if its input is greater than $n - 1$. Let $n - 1 = \tilde{x}_{\lceil\log_2(n)\rceil-1} \cdots \tilde{x}_0 = \tilde{x}$ and S be the set of indices of $\tilde{x}$ equal to zero, i.e. $S = \{j \mid \tilde{x}_j = 0\}$. Hence, the first penalty reads [67]:

$$C_1(x_i) = \sum_{j \in S} x_{i,j} \prod_{k=j+1}^{\lceil\log_2(n)\rceil-1} (1 - (x_{i,k} - \tilde{x}_k)^2). \quad (37)$$

The second subcost C_2 is used to penalize if two sequences x and y encode the same city [67]:

$$C_2(x, y) = \prod_{k=0}^{\lceil\log_2(n)\rceil-1} (1 - (x_k - y_k)^2). \quad (38)$$

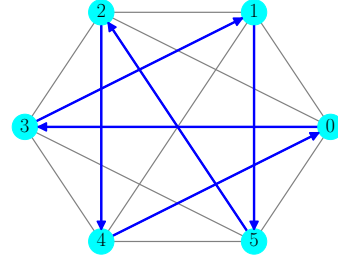


Figure 6: HUBO encoding of TSP with $n = 6$ cities labeled with integers from 0 to $n - 1$ on a complete graph. The encoding uses $n\lceil\log_2(n)\rceil = 18$ qubits, and an assignment is of the form $x = x_0x_1x_2x_3x_4x_5$. The blue route describes the visit of the cities in order: 0, 3, 1, 5, 2, 4. Its corresponding encoding is $x = 000011001101010100$.

5 Numerical results for SAMBA-GQW

5.1 Performance metrics

Throughout, we use several metrics for the evaluation of performance.

First, the quality of a produced quantum state $|\psi\rangle$ is evaluated with the expectation of its probability distribution as,

$$\mathbb{E}[\psi] = \sum_{x \in \{0,1\}^n} \mathbf{p}(x) \cdot \mathbf{q}(x), \quad (39)$$

where $\mathbf{p}(x) = |\langle x|\psi\rangle|^2$ is the probability of measuring basis state $|x\rangle$. The function $\mathbf{q}(x) \in [0, 1]^6$ outputs the quality of a decision x such that the optimal and worst decision respectively lead to $\mathbf{q}(x^*) = 1$ and $\mathbf{q}(\arg \max(C)) = 0$. In this context, Eq. (39) gives information about the distance between $|\psi\rangle$ and $\sum_{x^*} \alpha_{x^*} |x^*\rangle$ with $\sum_{x^*} |\alpha_{x^*}|^2 = 1$. Note that, if constraints are present, we put $\mathbf{q}(x) = 0$ for all unfeasible decisions ($\forall x \notin \mathcal{F}$).

We use then the normalized participation ratio (or PR for short) to quantify the localization of a state vector:

$$\text{PR}(\psi) = \left(2^n \sum_{x \in \{0,1\}^n} \mathbf{p}(x)^2 \right)^{-1}. \quad (40)$$

The uniform superposition leads to $\text{PR} = 1$ and the fully localized state gives $\text{PR} = 1/2^n$. This metric gives information on the sparsity of $|\psi\rangle$.

⁶Since the optimal solution is that minimizing the cost function, we define the quality function as the approximation ratio $\mathbf{q}(x) = \frac{C_{\max} - C(x)}{C_{\max} - C_{\min}}$.

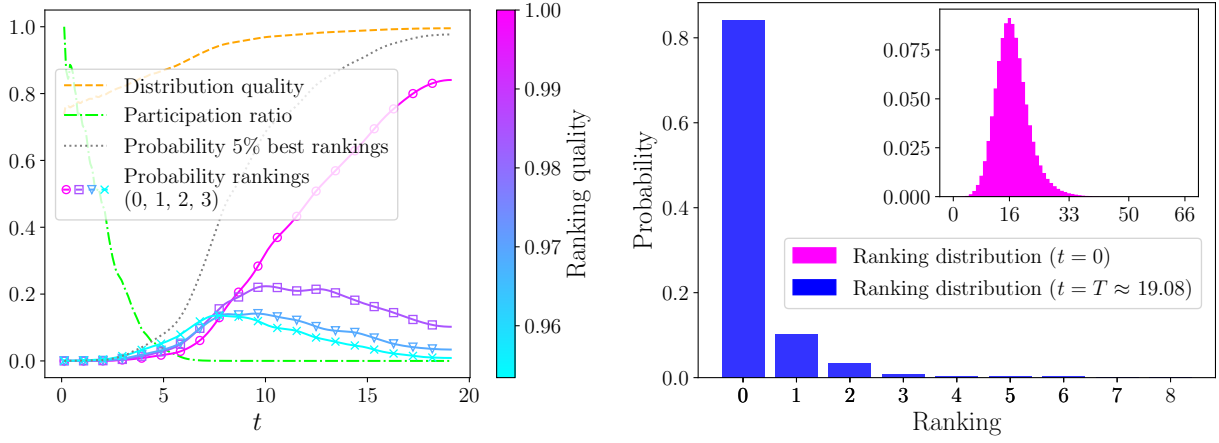


Figure 7: SAMBA-GQW Hamiltonian simulations for unweighted MaxCut on random Erdős-Rényi graphs for $n = 20$ qubits with a sampling of $q = n^2$ states.

Finally, we label the quality level of a decision x with its “ranking” $\mathbf{r}(x) \in \mathbb{N}$. Optimal solutions have a ranking of 0, second-best decisions have a ranking of 1, and so on. Let S_r be the set of decisions with ranking r , i.e. $S_r = \{x \in \{0,1\}^n | \mathbf{r}(x) = r\}$. The total number of rankings is equal to the number of distinct cost values, which corresponds to the dimension (cardinality) of the image of the cost function $C : \{0,1\}^n \rightarrow \mathbb{R}$. The probability of measuring a decision of ranking r reads:

$$\mathbb{P}_r[\psi] = \sum_{x \in S_r} |\langle x | \psi \rangle|^2. \quad (41)$$

5.2 Results

Hamiltonian simulations are performed for all considered problems, and quantum circuit simulations are additionally carried out for MaxCut, MIS, and LABS. Moreover, for all problems whose instance generation is parameterized (all except LABS) we generate 10 different instances each time, and we present the average of the results obtained. Across all Hamiltonian simulations, we construct the hopping rate Γ by sampling only $q = n^2$ states among 2^n .

In general, all the simulations consistently support SAMBA-GQW as a method that in a short evolution time can shift an initial state distribution into a very localized distribution around the optimal solution. The most probable ranking is often the optimal solution, or ranking $\ll 10$. Even then, the optimal solution is within the 5% best rankings and a classical post-processing can easily extract it. In this context, SAMBA-GQW

requires little tuning, is consistent, and scales reasonably well with the number of decisions.

5.2.1 MaxCut

We generate instances on Erdős-Rényi graphs [68] of $n = 20$ vertices, both weighted ($w_{ij} \in [-10, 10]$) and unweighted ($w_{ij} = 1$), with an edge appearance probability of $\frac{1}{2}$.

Hamiltonian simulations. We present the distribution quality, participation ratio, measurement probabilities per ranking over time and the probability distributions over rankings at $t = 0$ and $t = T$ on Fig. 7 (resp. 8) for unweighted (resp. weighted) MaxCut. At $t = 0$ we show the complete ranking distribution, however, at $t = T$ we only display rankings with measurement probability higher than 10^{-3} .

We obtain a high final success probability of measuring the optimal solution, i.e. ranking 0, since it is of order $\mathbb{P}[\psi_T] \approx 0.8$. In both unweighted and weighted scenario, the probability of measuring decisions with ranking among the 5% best is almost 1. Moreover, at $t = 0$ the quantum state is a uniform superposition among the 2^n computational basis states resulting in a participation ratio of $\text{PR}(\psi_0) = 1$. We observe that this ratio decreases rapidly to converge to $\text{PR}(\psi_T) \approx 1/2^n$, resulting in a more localized state. Lastly, we notice that the initial distribution quality is $\mathbb{E}[\psi_0] \approx 0.75$ for unweighted MaxCut and $\mathbb{E}[\psi_0] \approx 0.5$ for weighted MaxCut. This difference means that the unweighted case contains more high quality solutions than

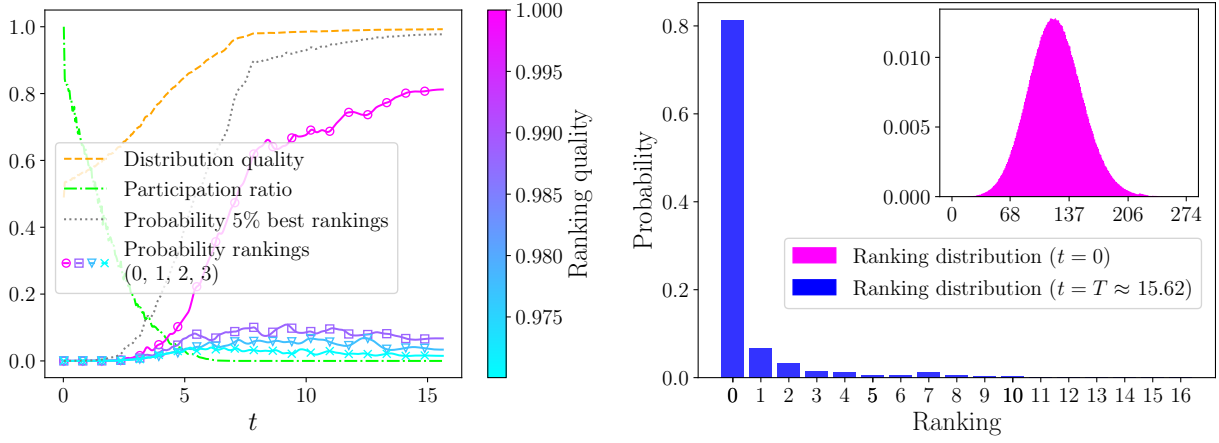


Figure 8: SAMBA-GQW Hamiltonian simulations for weighted MaxCut on random Erdős-Rényi graphs for $n = 20$ qubits with a sampling of $q = n^2$ states.

weighted MaxCut. We notice that the time evolution is fairly short as it reaches $T \approx 20$ (resp. $T \approx 16$) for unweighted (resp. weighted) MaxCut, this time evolution difference is caused by lower energy gaps in weighted instances. In both cases, we observe an initial Gaussian ranking distribution, although it is smoother for weighted MaxCut since this problem contains 275 different rankings compared with 67 for unweighted MaxCut. Furthermore, as indicated by the distribution quality, the initial distribution for the unweighted case is shifted towards higher quality decisions, while the weighted Gaussian is centered around the middle rankings. After the evolution induced by SAMBA-GQW, the distribution is almost fully localized on the optimal solutions for both cases.

Quantum circuit simulations. We run quantum circuit simulations for even system sizes $n \in [10, 30]$ and for different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. For each sampling value, the hopping-rate schedule is discretized using various values of p_* , where p_* denotes the number of cuts uniformly applied to each linear subinterval of the hopping rate. Therefore, the resulting quantum circuit has a total of $\bar{p} = |\mathcal{E}| \cdot p_*$ cost and mixer Hamiltonian layers in the circuit, where $|\mathcal{E}| \leq q$ represents the number of distinct sampled energies, i.e., the number of interpolation points used to construct the hopping rate (excluding the additional zero-energy point).

Detailed results on unweighted MaxCut for $n = 30$ qubits are presented in Fig. 9. For sampling values $q < n^3$, the number of distinct sampled

energies remains small, with $|\mathcal{E}| < 5$, whereas for $q = n^3$, it increases to a range between 10 and 30. This directly impacts performance: the highest approximation ratios, close to 1, are achieved for $q = n^3$, making it the best-performing sampling value among those considered. We also observe a clear trade-off between solution quality and number of circuit layers, as the choice $q = n^3$ leads to the largest circuits. We show the approximation ratio and number of layers in the circuit as a function of the number of qubits n and the discretization parameter p_* in Fig. 10. The approximation ratio exhibits a clear threshold behavior as a function of the number of circuit layers p_* . For all sampling choices, the performance is relatively poor at very small depth ($p \leq 2$), with the lowest values occurring for small system sizes n , and then increases sharply around $p_* \approx 3$, beyond this point, the ratio remains high over a broad range of n and improves further as q is increased. In particular, for $q = 5n$ and $q = 2n \log_2 n$, a pronounced degradation appears again at large n , whereas for $q = n^2$ this deterioration is pushed to the largest system sizes, and for $q = n^3$ it is essentially absent, with the approximation ratio staying close to its optimal plateau almost uniformly in n once $p_* \geq 3$. The layer counts show a complementary pattern. They are small at low depth, then grow rapidly with p_* , with the growth becoming much stronger as q increases. For $q = 5n$ and $q = 2n \log_2 n$, the required number of layers remains moderate and even decreases for the largest n , while for $q = n^2$ and especially $q = n^3$ the high-performance region is associated with a substantial overhead, reaching the largest layer

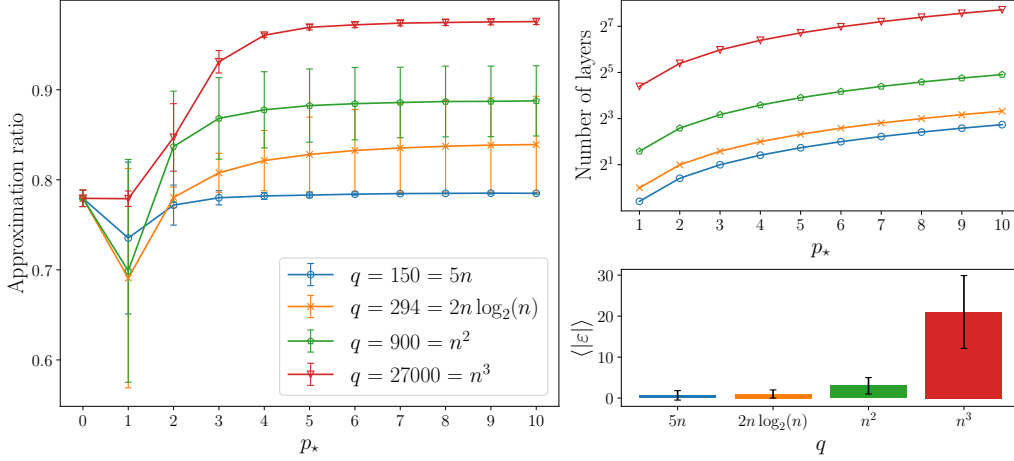


Figure 9: SAMBA-GQW quantum circuit simulations for unweighted MaxCut for $n = 30$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The approximation ratio is shown as a function of the discretization parameter $p_* \in [0, 10]$ ($p_* = 0$ corresponds to the circuit just composed of a single Hadamard layer), indicating the size of the subintervals in which we split the subintervals uniformly, i.e. $p_l = p_*$, see Fig. 2 (left). The number of layers in the circuit, in which one unit corresponds to the black box application of cost and mixer Hamiltonians (top right). The amount of different sampled energies $|\mathcal{E}|$ as a function of the number of samples q (bottom right).

counts at large p_* . Overall, increasing q stabilizes high approximation ratios across system sizes, but at the cost of deeper circuits.

We now present the corresponding figures for the weighted MaxCut problem. From Fig. 11, we observe that the number of distinct sampled energies increases smoothly with the sampling parameter q . Interestingly, at $p_* = 10$, the best sampling strategy on average is $q = 5n$. This indicates that, for this problem, such a sampling level provides a sufficient number of sampled energies, whereas larger sampling values require deeper circuits to yield performance improvements. This behavior is further confirmed in Appendix D, where larger circuits with depths up to $p_* = 30$ are considered. Figure 12 shows that, in contrast to the unweighted case, the weighted MaxCut problem exhibits a smoother and size-dependent improvement with circuit depth. Specifically, the depth required to achieve a given approximation ratio increases with the number of qubits. Increasing the sampling parameter q can partially compensate for this complexity by shifting the onset of high-performance regimes toward smaller depths and enabling approximation ratios above 0.9 for most system sizes. Nevertheless, these gains are accompanied by a pronounced increase in circuit depth, with the number of layers scaling steeply with both p_* and q , reaching several thousand layers for the largest sampling values. This behavior highlights a fundamental trade-off:

larger q improves robustness and solution quality but substantially increases the required computational resources.

5.2.2 Maximum independent set (MIS)

In the case of unweighted MIS, we generate Erdős-Rényi graphs [68] of $n = 20$ vertices with an edge appearance probability of $1/2$, and set $w_i = 1$ for all the vertices. Regarding UD-MIS, we generate UD graphs of $n = 20$ vertices and we select the weights in the spirit of Ref. [58] such that:

$$w_i = 1 + \frac{10 \cdot C_D(i)}{\max_{j \in V} C_D(j)}, \quad (42)$$

with $C_D(i) = \deg(i)/(|V| - 1)$ the degree centrality of vertex i . Furthermore, the procedure we use to generate UD graphs is detailed in the Appendix D of Ref. [58].

Hamiltonian simulations. We present the quality distribution, participation ratio and probabilities per ranking over time for random Erdős-Rényi (resp. UD) graphs on Fig. 13 (resp. 14). In the case of random unweighted graphs, the highest measurement probability is that of ranking 3 with $\mathbb{P}_3[\psi_T] \approx 0.15$, which has a quality close to 0.99. We see that optimal solutions have a relevant success probability $\mathbb{P}_0[\psi_T] \approx 0.02$, considering that the evolution starts in a uniform

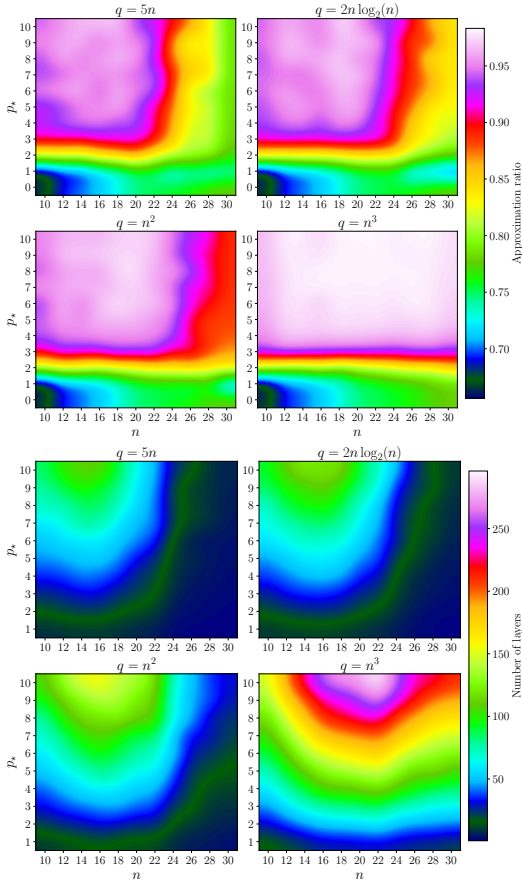


Figure 10: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 30]$ and the discretization parameter $p_* \in [1, 10]$ for unweighted MaxCut with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. We only include $p_* = 0$ for the approximation ratio to show the initial approximation ratio when the circuit only contains a Hadamard layer.

superposition of 2^n states. Moreover, the probability of measuring decisions whose ranking is in the top 5% of all rankings is around 0.9.

Regarding UD graphs, optimal solutions have the highest measurement probability with $\mathbb{P}_0[\psi_T] \approx 0.25$. Furthermore, the probability of measuring decisions with the top 5% rankings almost reaches 1. In both cases, the participation ratio decreases rapidly, indicating that the quantum state goes from a uniform superposition to a more localized state. Additionally, both random and UD graphs have a high initial distribution quality $\mathbb{E}[\psi_0] \approx 0.8$, suggesting that there are more high quality decisions than low quality ones. Lastly, we note that time evolution is extremely short since it reaches $T \approx 6$ (resp. $T \approx 7$) for random (resp. UD) graphs. Moreover, we see that the ranking distribution is shifted to the left in

both cases. However, for UD graphs, the number of different rankings is significantly higher than for random graphs, with 1897 against 246. In addition, the distribution for UD graphs seems to be composed of two Gaussians instead of one for random graphs. In both cases, the evolution induced by SAMBA-GQW shrinks the initial distribution to a distribution localized around high quality decisions, with high success probability on optimal solutions for UD graphs.

In addition, we also execute SAMBA-GQW on two instances proposed in Ref. [69], *farm* and *mammalia-kangaroo-interactions* [70] of $n = 17$ vertices with $\lambda = 2$ and $w_i = 1$. We display these instances on Fig. 15.

We obtain similar results for the two instances, we see on Fig. 16 (resp. 17) for *farm* (resp. *mammalia-kangaroo-interactions*) that the highest measurement probability is that of the optimal solutions in both cases, with $\mathbb{P}_0[\psi_T] \approx 0.5$ (resp. $\mathbb{P}_0[\psi_T] \approx 0.45$). For *farm*, the probability of measuring decisions with ranking in top 5% of all rankings reaches approximatively 0.96, and 1 for *mammalia-kangaroo-interactions*. The participation ratio decreases up to a localized state quickly and distribution quality is initially high with $\mathbb{E}[\psi_0] \approx 0.8$ for the two instances. The required evolution time is very short since it reaches $T \approx 10$ in both cases. Furthermore, we see that the Gaussian ranking distributions are shifted to higher quality decisions, as suggested by the distribution quality, and instance *mammalia-kangaroo-interactions* owns more different rankings than *farm* with 158 against 71 rankings. After the evolution induced by SAMBA-GQW, the ranking distributions are almost fully localized on optimal solutions. Moreover, in both cases only six rankings have a measurement probability higher than 10^{-3} .

Quantum circuit simulations. For unweighted MIS, the results for $n = 30$ are presented in Fig. 18. The approximation ratio rapidly converges to a plateau for $p_* \geq 2$, indicating that shallow circuits are sufficient to achieve near-optimal solutions. In parallel, the number of layers increases steadily with p_* and is strongly amplified by larger sampling rates q . A broader perspective is provided in Fig. 19, where the algorithm exhibits a distinctive and highly favorable performance profile. Unlike MaxCut, the

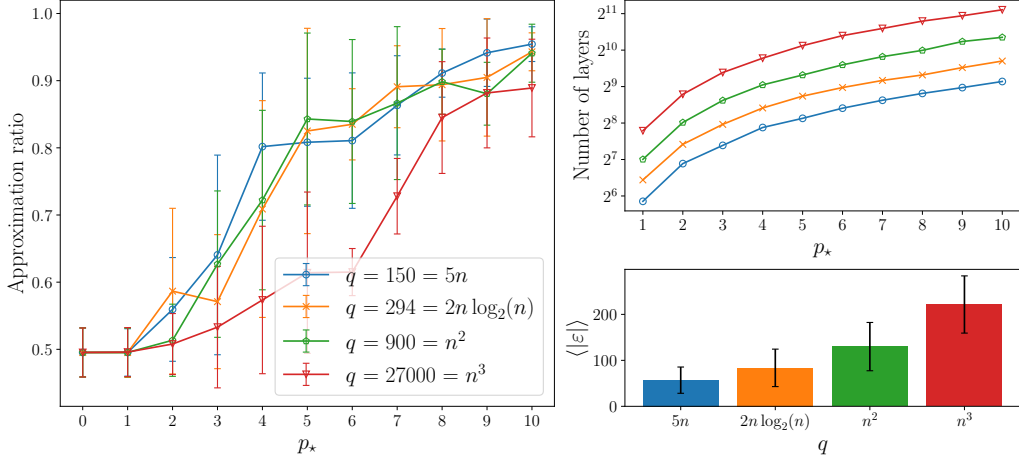


Figure 11: SAMBA-GQW quantum circuit simulations for weighted MaxCut for $n = 30$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

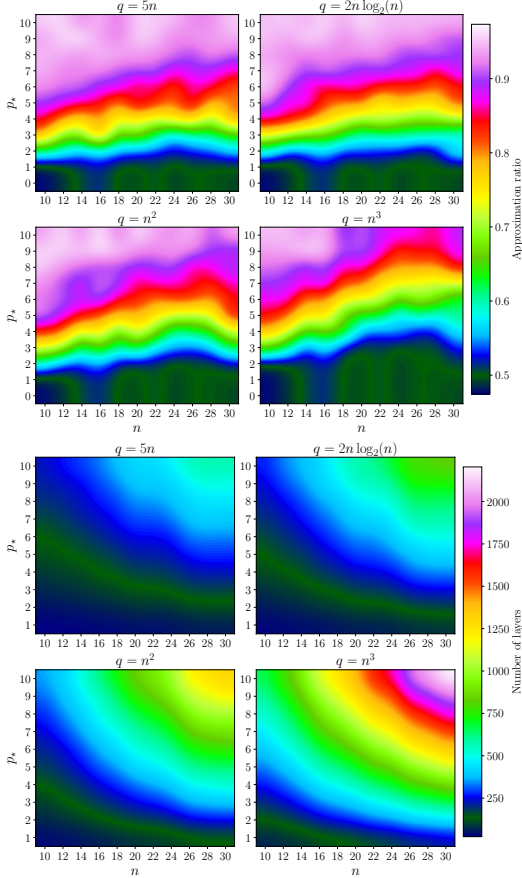


Figure 12: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 30]$ and the discretization parameter $p_* \in [1, 10]$ for unweighted MaxCut on random Erdős-Rényi graphs with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

approximation ratio displays a pronounced non-monotonic dependence on the circuit depth p_* ,

characterized by a systematic dip at $p_* = 1$ followed by a sharp recovery around $p_* \simeq 2$. Beyond this point, the performance rapidly saturates close to optimal values and remains largely independent of the system size n . This early saturation suggests that the essential combinatorial structure of unweighted MIS instances can be captured with very shallow circuits. Increasing the sampling parameter q primarily enhances the robustness of this high-performance regime rather than significantly improving its asymptotic value, with $q = n^3$ yielding near-optimal approximation ratios across all considered system sizes. The required resources, however, scale steeply with both p_* and q . The number of circuit layers grows rapidly with the circuit depth and system size, and increases dramatically as q is enlarged, reaching several thousand layers for the largest instances. Consequently, while larger values of q provide only marginal improvements in solution quality, they incur a substantial overhead in circuit complexity, highlighting a clear trade-off between robustness and computational cost.

We now present the results for UD-MIS. Looking at Fig. 20, we observe that the number of distinct sampled energies is significantly higher than in the unweighted MaxCut case on $n = 30$ qubits, reaching $|\mathcal{E}| = 2000$ for $q = n^3$. Moreover, sampling values $q \in \{5n, 2n \log_2(n), n^2\}$ yield approximately the same approximation ratio, whereas $q = n^3$ requires larger values of p_* to achieve a noticeable improvement. We also observe that at $p_* = 10$, all sampling strategies attain similar approximation ratios, with $q = n^3$ not provid-

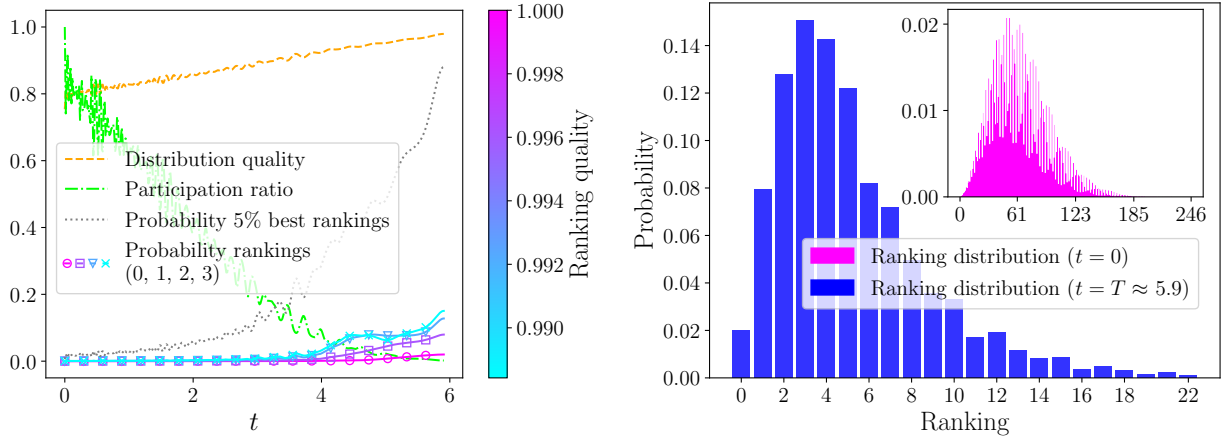


Figure 13: SAMBA-GQW Hamiltonian simulations for unweighted MIS on random Erdős-Rényi graphs MIS for $n = 20$ qubits with a sampling of $q = n^2$ states.

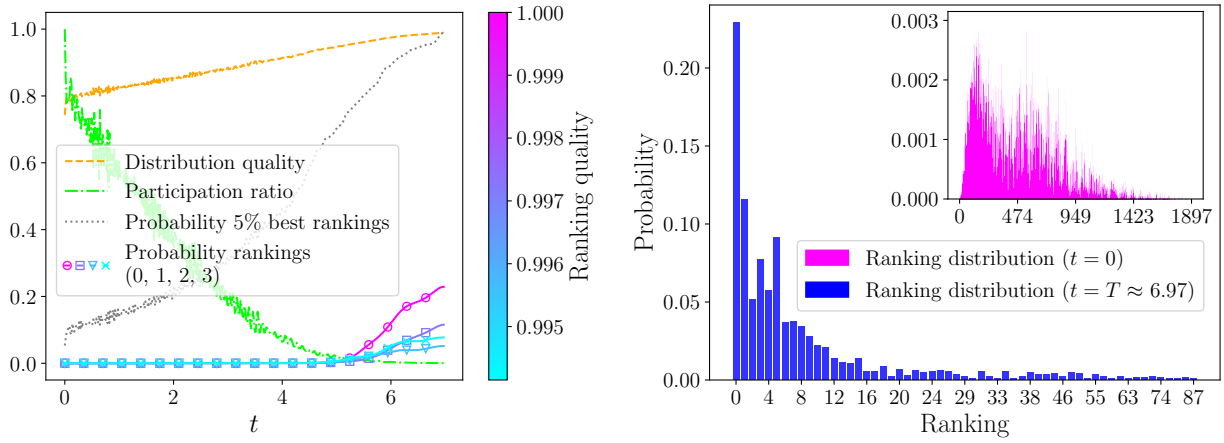


Figure 14: SAMBA-GQW Hamiltonian simulations for weighted MIS on Unit-Disk graphs MIS for $n = 20$ qubits with a sampling of $q = n^2$ states.

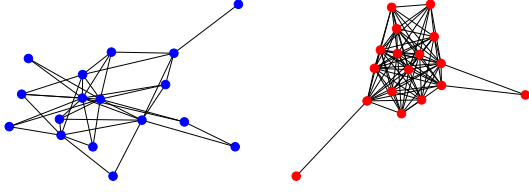


Figure 15: Instances *farm* (left) and *mammalia-kangaroo-interactions* (right) of $n = 17$ vertices.

ing any advantage. Considering the number of layers in the circuit, selecting $q = n^3$ is therefore the least favorable strategy, as the approximation ratio converges more slowly while leading to the highest circuit depth. These results highlight a clear trade-off between solution quality and circuit depth: higher sampling values require significantly deeper circuits without necessarily providing better performance. Figure 21 shows that the number of circuit layers increases rapidly with both the discretization parameter p_* and the sampling value q . In particular, for $q = n^3$, the circuit depth grows steeply with the system size, reaching several tens of thousands of layers for the largest instances, whereas smaller sampling values ($q = 5n$ and $q = 2n \log_2 n$) lead to significantly shallower circuits. Furthermore, all sampling strategies achieve high approximation ratios at relatively small discretization parameter values ($p_* \approx 3-4$), with only marginal improvements for larger p_* . Increasing the sampling parameter to $q = n^3$ does not yield a substantial performance advantage and requires considerably deeper circuits to reach comparable approximation ratios. These results highlight a clear trade-off between solution quality and circuit depth. Moderate sampling values, such as $q = 5n$ or $q = 2n \log_2 n$, provide the most favorable balance, achieving near-optimal approximation ratios while keeping the circuit depth manageable. In contrast, larger sampling values significantly increase the required computational resources without delivering commensurate gains in performance.

5.2.3 Portfolio optimization

We use the file JKMP1 500_1 of dataset [71] to generate 10 instances of $n = 20$ assets each. Since this file contains portfolio information of 500 variables, we generate different instances by randomly drawing $n = 20$ variables. We set the

risk appetite to $\lambda = 1/2$ in Eq. (23) and the number of assets to $k = n/2 = 10$. Therefore, the initial state is the uniform superposition over basis states of Hamming weight $k = 10$. Since we use XY -mixers, only the feasible space is explored by the walker, i.e., only the subgraph of Hamming weight $k = 10$ is explored. The feasible space contains $\binom{n=20}{k=10} = 184756$ basis states linked with ring connectivity. In the sampling protocol, we do not restrict ourselves to feasible solutions. Instead, we randomly select any possible solution from the full 2^n solution space, including infeasible ones. Additionally, for performance evaluation, non-feasible decisions, i.e., states with Hamming weights not equal to k , have a quality of 0.

We see on Fig. 22 that the probability of measuring the optimal solution is the highest at the end of the evolution with $\mathbb{P}_0[\psi_T] \approx 0.06$, followed closely by ranking 1 decision with $\mathbb{P}_1[\psi_T] \approx 0.05$. Moreover, the probability of measuring decisions with ranking in the top 5% of all rankings reaches approximatively 0.85. The initial distribution quality is $\mathbb{E}[\psi_0] \approx 0.5$, which indicates a good balance between high and low quality solutions. Furthermore, we observe that the time evolution $T \approx 28056$ is significantly higher than it is for MaxCut and MIS since the energy gaps are significantly lower for the portfolio optimization instances, leading to longer transfer time. We can also observe that the initial ranking distribution is uniform for portfolio optimization with 184756 different rankings for the feasible decisions. The evolution induced by SAMBA-GQW leads to a high quality distribution where the first two highest measurement probabilities respectively correspond to ranking 0 and 1.

5.2.4 LABS

Hamiltonian simulations. We present the results for LABS on $n = 20$ qubits, where we minimize the energy (instead of maximizing the merit factor). As seen on Fig. 23, the initial distribution quality is very high with $\mathbb{E}[\psi_0] \approx 0.95$. This phenomenon indicates that most of the 2^n states have a very high quality and may impact the difficulty of solving the problem since it is easy to be stuck on local minima close to the optimal solutions. We observe that after the short time evolution $T \approx 1.4$, the optimal solutions do not have the highest measurement probabilities. However, the probability of measuring

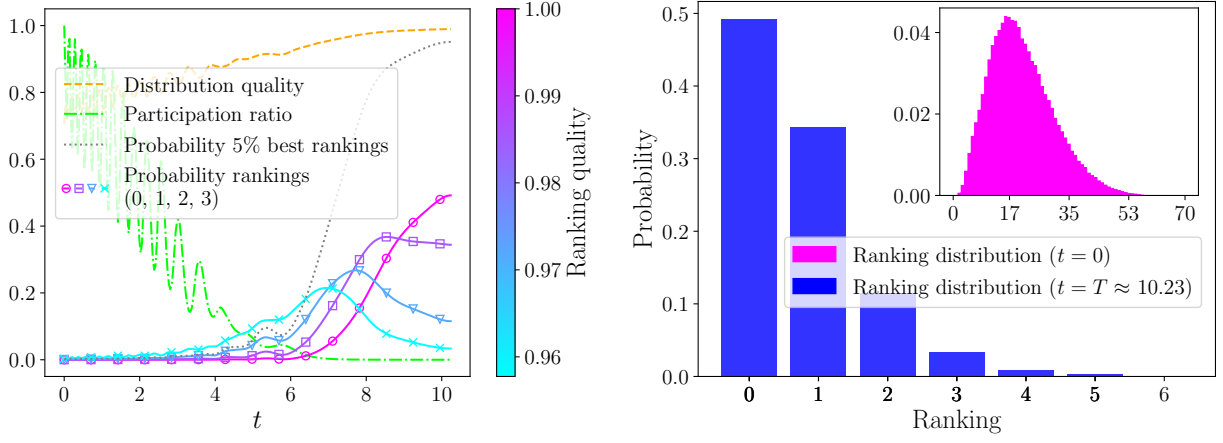


Figure 16: SAMBA-GQW Hamiltonian simulations for MIS on *farm* instance from Refs. [69, 70], for $n = 17$ qubits with a sampling of $q = n^2$ states.

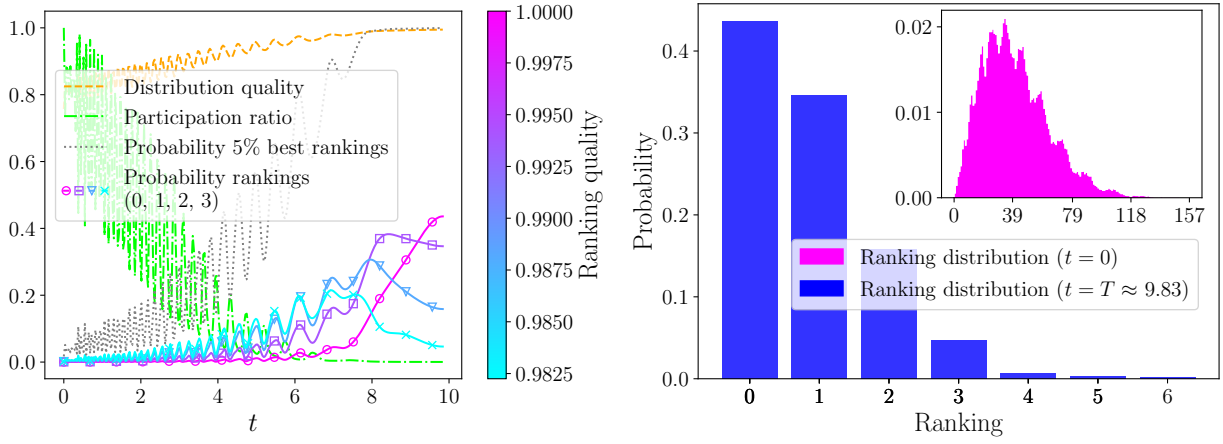


Figure 17: SAMBA-GQW Hamiltonian simulations for MIS on *mammalia-kangaroo-interactions* instance from Refs. [69, 70], for $n = 17$ qubits with a sampling of $q = n^2$ states.

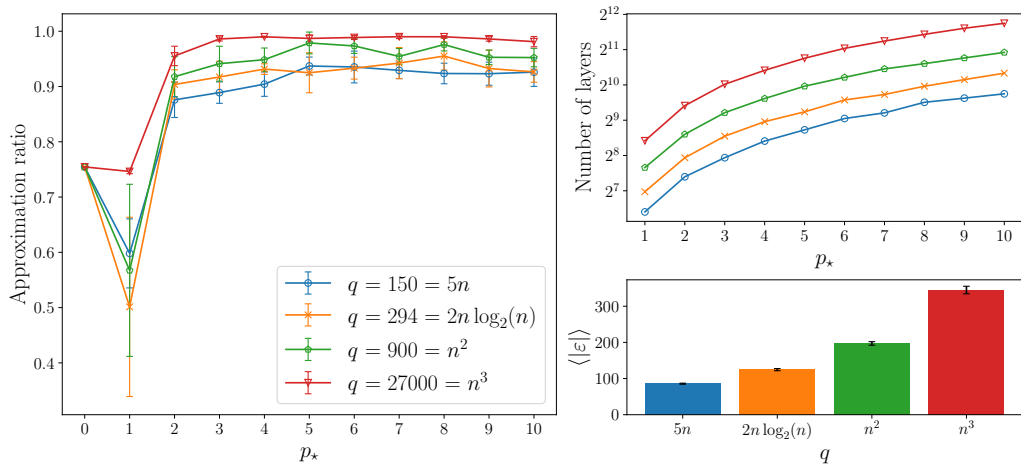


Figure 18: SAMBA-GQW quantum circuit simulations for unweighted MIS on random Erdős-Rényi graphs for $n = 30$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

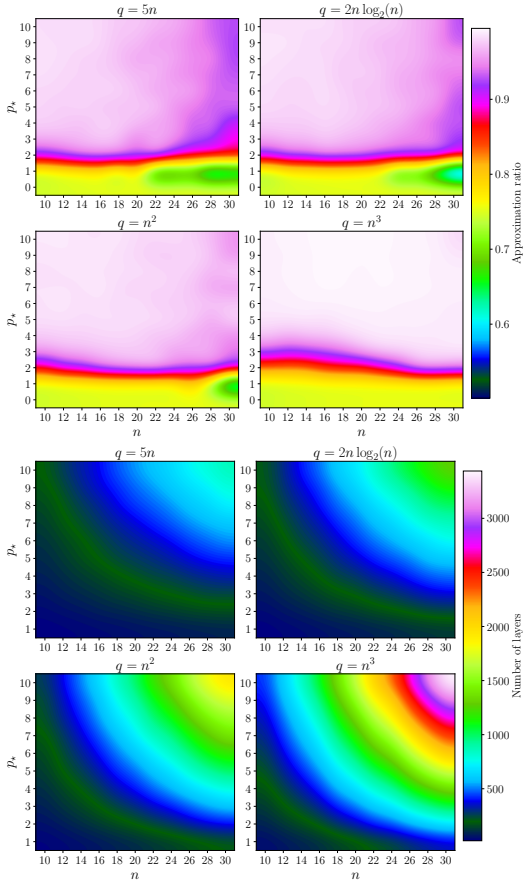


Figure 19: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 30]$ and the discretization parameter $p_* \in [1, 10]$ for unweighted MIS on random Erdős-Rényi graphs with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

states with ranking in the top 5% of all rankings almost reaches 1, which indicates good performance of SAMBA-GQW since the algorithm finds almost optimal solutions with high probabilities by only sampling $q = n^2$ states. Note that, in general, even restricting the search to the top 5% of rankings may still correspond to an exponentially large search space. However, in the present case, the participation ratio indicates that the quantum state converges toward a well-localized distribution, suggesting that only a small subset of configurations carries a significant fraction of the probability mass. Lastly, the participation ratio decreases rapidly as the quantum state localizes during the evolution. We observe this behavior on Fig. 23 since only 18 rankings among 254 have a measurement probability higher than 10^{-3} after the evolution induced by the algorithm. The Gaussian ranking distribution is strongly shifted to low rankings, i.e. high quality decisions, as suggested by the initial distribution quality. Furthermore, after the evolution, four high quality rankings have a measurement probability higher than 0.1 with $\mathbb{P}_6[\psi_T] \approx 0.24$, $\mathbb{P}_7[\psi_T] \approx 0.21$, $\mathbb{P}_5[\psi_T] \approx 0.18$ and $\mathbb{P}_4[\psi_T] \approx 0.12$. The initial measurement probabilities of these rankings are respectively 0.0008, 0.0017, 0.0004, and 0.0001, making the final probabilities a sign of good performance of SAMBA-GQW. The cumulated probability of measuring rankings $r \in \{4, 5, 6, 7\}$ is very high since it reaches 0.75. The optimal solutions have a significantly lower measurement probability with $\mathbb{P}_0[\psi_T] \approx 0.005$, however at $t = 0$ this probability is $\mathbb{P}_0[\psi_0] \approx 7 \times 10^{-6}$. Thus, it is fair to consider the final measurement probability a sign a good performance of the algorithm since it multiplied the initial optimal probability by approximately 656 by sampling only $q = n^2$ states among 2^n . Moreover, since the quantum state is almost localized at $t = T$ since only 18 rankings among 254 have a measurement probability higher than 10^{-3} , including optimal solutions, one can perform several measurements in the computational basis and classically select the state with lowest energy. This way, one can increase the likeliness of recovering optimal solutions.

Quantum circuit simulations. We present the results for LABS on $n = 30$ qubits in Fig. 24. We observe that only the cubic sam-

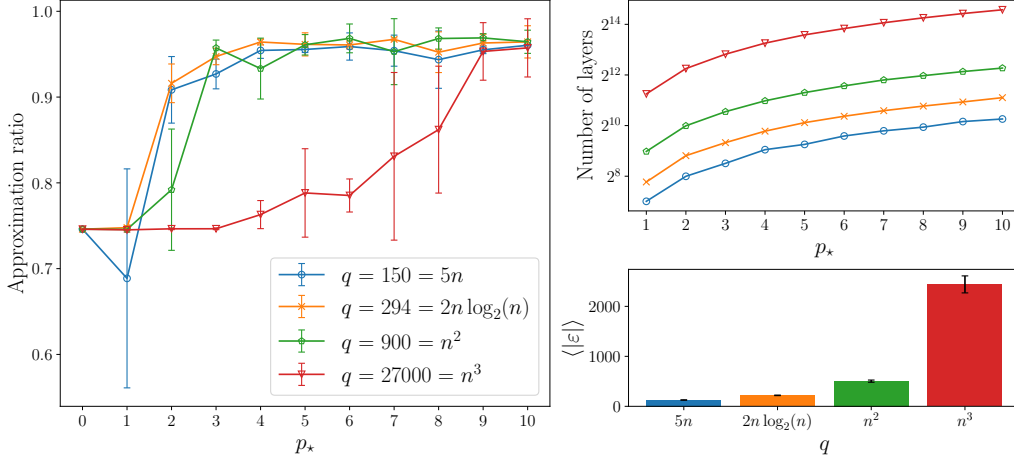


Figure 20: SAMBA-GQW quantum circuit simulations for weighted MIS on unit-disk graphs for $n = 30$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

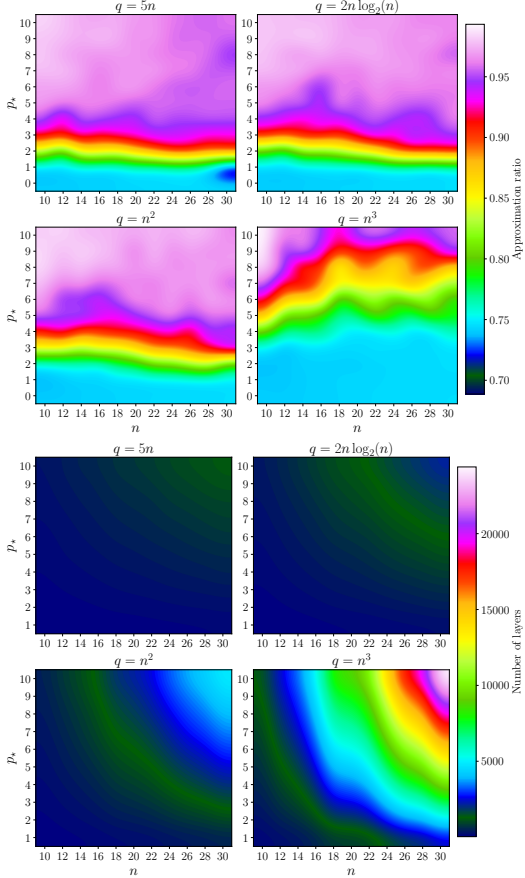


Figure 21: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 30]$ and the discretization parameter $p_* \in [1, 10]$ for weighted MIS on unit-disk graphs with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

pling regime, $q = n^3$, leads to variations in the approximation ratio as the circuit depth increases. This is explained by the fact that, for $q \in \{5n, 2n \log_2(n), n^2\}$, no energy gaps were sampled, resulting in a quantum circuit composed only of Hadamard gates applied to each qubit. In contrast, the protocol with $q = n^3$ successfully extracted four energy gaps. This behavior arises because the sampling protocol (see Algorithm 2) considers only positive energy gaps, as shown in Appendix E, taking the absolute values of the gaps eliminates this zero-sampled-energy phenomenon. Moreover, we illustrate the scaling of the number of sampled energy gaps as a function of n in Appendix C for MaxCut, MIS and LABS. For the case $q = n^3$, we observe a slight improvement in the approximation ratio with increasing circuit depth. We also note that the initial approximation ratio is already very high, since the LABS problem possesses a large proportion of high-quality solutions. From Fig. 25, we observe that increasing the number of samples becomes necessary to improve the approximation ratio as n grows.

5.2.5 MAX- k -SAT

We generate random instances of MAX- k -SAT by randomly drawing m clauses uniformly from the set of all possible clauses $\Phi_{k\text{-SAT}}$ with $|\Phi_{k\text{-SAT}}| = 2^k \binom{n}{k}$. The hardness of MAX- k -SAT strongly depends on the clause-to-variable ratio $\alpha = m/n$. There exists a threshold α_c , in which no known polynomial-time algorithm can produce an ϵ -approximate solution with high prob-

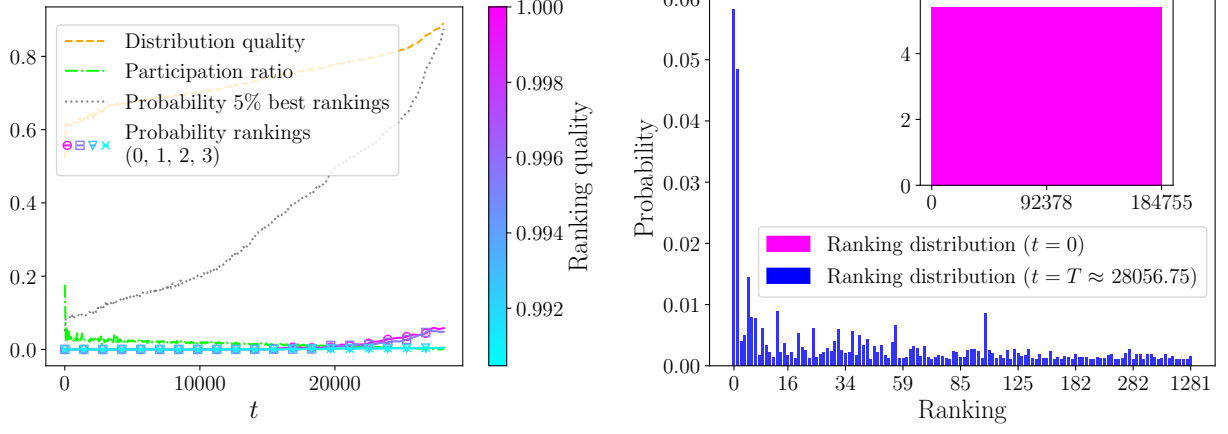


Figure 22: SAMBA-GQW Hamiltonian simulations on portfolio-optimization with XY -mixers (with ring connectivity) for $n = 20$ qubits with a sampling of $q = n^2$ states. The instances are from dataset of Ref. [71].

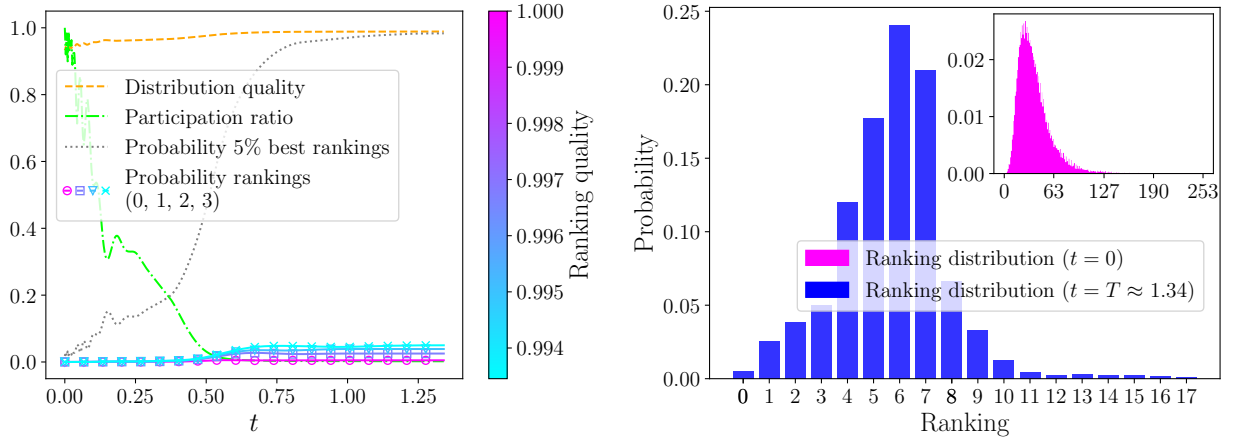


Figure 23: SAMBA-GQW Hamiltonian simulations on LABS for $n = 20$ qubits with a sampling of $q = n^2$ states.

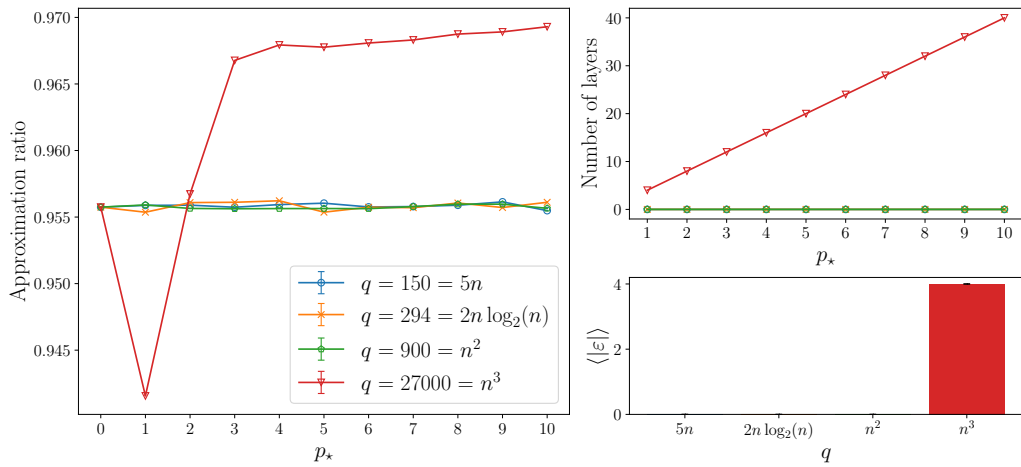


Figure 24: SAMBA-GQW quantum circuit simulations for LABS for $n = 30$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

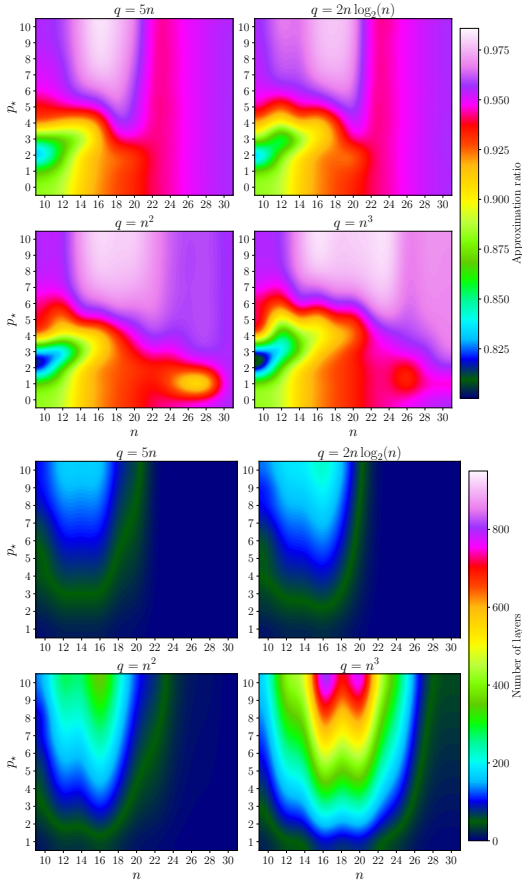


Figure 25: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 30]$ and the discretization parameter $p_* \in [1, 10]$ for LABS with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

ability [73]. This threshold is hard to determine for any value of k but can be estimated using cavity method [74, 75]. In practice, we run experiments for $k \in [3, 10]$ and we set the number of clauses to $m = \lfloor \alpha_k n \rfloor$ where the values of $\alpha_k \equiv \alpha_c$ are shown in Table 10 of Ref. [72]. We present the results for MAX- k -SAT on $n = 20$ qubits for degrees $k \in \{3, 4, 5, 6, 7, 8, 9, 10\}$ on Fig. 26. We observe that the evolution induced by SAMBA-GQW leads to a measurement of optimal solutions with highest probabilities only for $k \in \{3, 4\}$. As k increases, optimal solutions still have a relevant measurement probability, and in the worst case the highest measurement probability is that of ranking 2, which indicates good performance. We notice that no matter the degree k , distribution quality starts around $\mathbb{E}[\psi_0] \approx 0.6$, and it reaches lower values $\mathbb{E}[\psi_T]$ as k increases, which translates an increase in the problem difficulty. We observe that participation ratio decreases less rapidly as k increases, indicating that the state vector requires more evolution time to reach a more localized state with the increase of problem's degree. We notice that evolution time T does not seem to be impacted by the degree k since it is approximatively short and constant with $T \approx 20$. This observation indicates that instances of different degree have energy gaps of similar magnitude. Looking at the initial Gaussian ranking distribution, we see that they are slightly shifted to the left towards higher quality decisions. Furthermore, the number of different rankings does not exceed 38 which is quite low considering that we work on instances with 2^{20} decisions.

The probability of measuring decisions in the top 10% of rankings, reaches high values close to 1 for $k \in \{3, 4, 5, 6, 7\}$. We observe a slight drop in performance for $k \in \{8, 9, 10\}$, since their respective measurement probabilities of the 10% rankings are approximatively 0.8, 0.7, and 0.55. However, since the number of different rankings does not exceed 38, measuring the top 10% of rankings is approximatively equivalent to only measuring the first four rankings. Thus, even if this probability decreases with the increases of the problem's degree k , the performance of SamBa are still great considering that in the worst case for $k = 10$, the highest measurement probability is that of ranking 2 with the highest probability $\mathbb{P}_2[\psi_T] \approx 0.32$. For every degree k , we observe

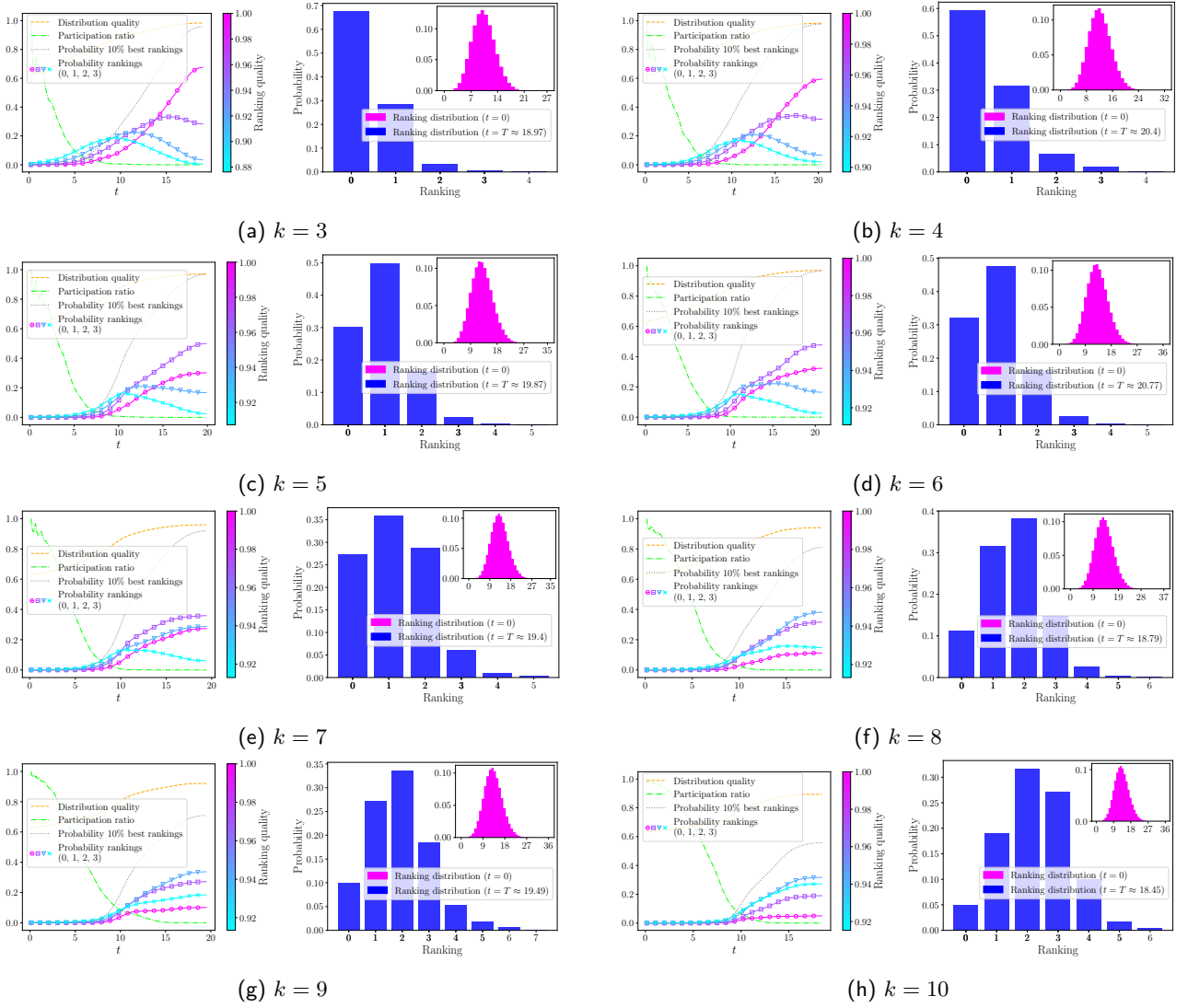


Figure 26: SAMBA-GQW Hamiltonian simulations on MAX- k -SAT for $n = 20$ qubits with a sampling of $q = n^2$ states. We present the results for degrees $k \in \{3, 4, 5, 6, 7, 8, 9, 10\}$, where the degree corresponds to the number of literal per clause. The number of clause is set to $m = \lfloor \alpha_k n \rfloor$ where the values of α_k are shown in Table 10 of Ref. [72].

that a sampling of only $q = n^2$ states for $n = 20$ is enough to guide the walker properly to high quality decisions, and even directly to optimal solutions for $k \in \{3, 4\}$. As the degree of the problem increases, it becomes harder to find optimal solutions, however, we still measure almost-optimal decisions with high probabilities. In such cases of higher k , optimal solutions could be recovered by performing several measurements and by efficiently selecting classically the best measured decision.

5.2.6 Travelling Salesperson Problem

We run TSP on complete graphs of $m = 6$ cities (vertices), therefore we use $n = m \lceil \log_2(m) \rceil = 18$

qubits for the HUBO encoding. We generate instances with long ($1 \leq w_{ij} \leq 100$) and short ($0 \leq w_{ij} \leq 1$) distances between the cities. This difference has a direct impact on the values of the energy gaps. With long distances we have short evolution times since gaps are quite high and with short distances the gaps get a lot smaller making the evolution time longer. We set the parameters of Eq. (36) to $\mu = 1$ and $\lambda = \gamma = 2\mu \max_{(i,j) \in E} w_{ij}$. We obtain similar results for long and short distance instances with a significant difference in the evolution time as shown on Figs. 27 and 28. The long distance instances require an evolution time $T \approx 1$ against $T \approx 80$ for the short distance instances. This comes from

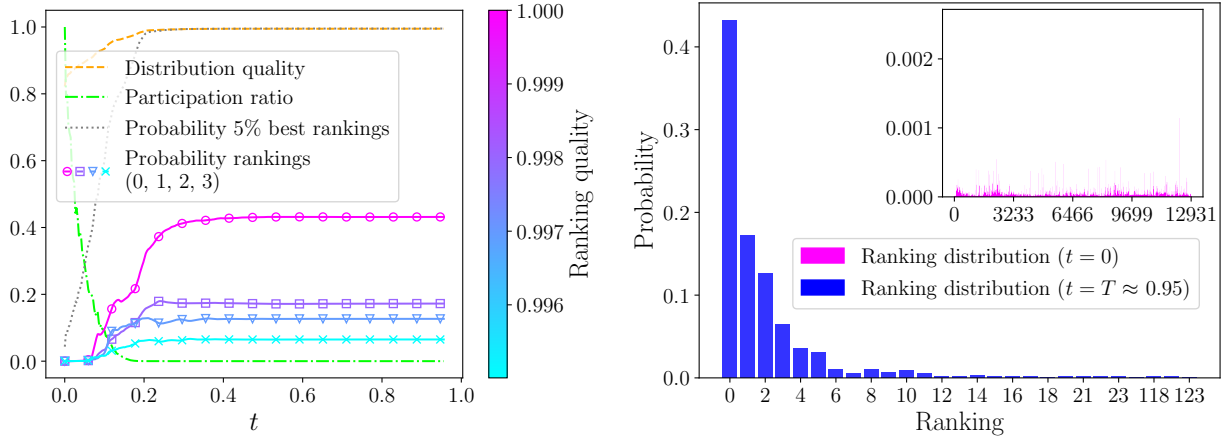


Figure 27: SAMBA-GQW Hamiltonian simulations on complete graphs with weight bounded by $1 \leq w_{ij} \leq 100$ distances on TSP for $n = 18$ qubits with a sampling of $q = n^2$ states.

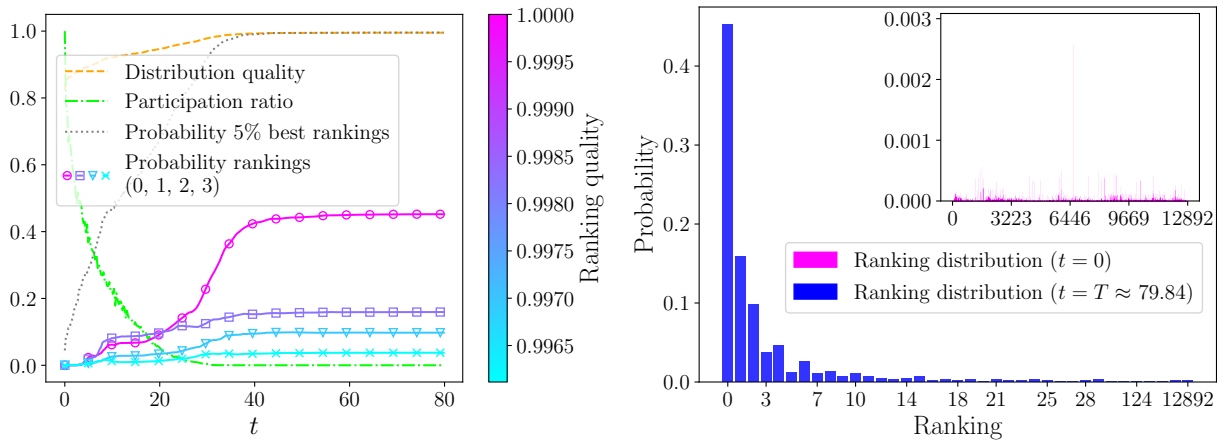


Figure 28: SAMBA-GQW Hamiltonian simulations on complete graphs with weight bounded by $0 \leq w_{ij} \leq 1$ distances on TSP for $n = 18$ qubits with a sampling of $q = n^2$ states.

the difference in energy gaps which are lower in the second case, leading to longer evolution time. For both long and short distance instances, the optimal solutions have the highest measurements probabilities, reaching $\mathbb{P}_0[\psi_T] \approx 0.4$ with slightly better results for short distances. The probability of measuring the top 5% rankings reaches 1 at $t \approx 0.2$ (resp. $t \approx 40$) for long (resp. short) distance instances, and the participation ratio decreases fast to approach $1/2^n$, indicating a localized state early in the evolution. The initial ranking distribution shown on Figs. 27 and 28 are spread among a higher amount of rankings than the previous studied problems. Long (resp. short) distance instances have 12932 (resp. 12893) different rankings, and the evolution induced by SAMBA-GQW leads the walker to the optimal solutions with high success probability even if the TSP problem is encoded as a HUBO.

6 Comparison with the state of the art

6.1 Guided quantum walk

The authors of Ref. [40] introduced the guided quantum walk (GQW) as an heuristic for combinatorial optimization. They follow the time evolution of Eq. (6) and use a classical optimizer to learn the optimal hopping rate Γ . Knowing that this function must be monotonically decreasing and positive [40], they reduce the number of learning parameters by using cubic Bézier curves. Such curves are defined by four control points $p_i = (x_i, y_i)^\top$ with $i \in \{0, 1, 2, 3\}$. Their coordinates reads:

$$\begin{aligned} x(\tau) &= x_0(1-\tau)^3 + 3x_1(1-\tau)^2\tau + 3x_2(1-\tau)\tau^2 \\ &\quad + x_3\tau^3 \\ y(\tau) &= y_0(1-\tau)^3 + 3y_1(1-\tau)^2\tau + 3y_2(1-\tau)\tau^2 \\ &\quad + y_3\tau^3. \end{aligned} \quad (43)$$

To ensure strict decrease conditions, they set $p_0 = (0, 1)$ and $p_1 = (1, 0)$, i.e. $x_0 = y_3 = 0$ and $y_0 = x_3 = 1$. Moreover, they define the hopping rate function as:

$$\Gamma(t) = y(t) \cdot 10^{a \cdot \alpha} + (1 - y(t)) \cdot 10^{b \cdot \beta}, \quad (44)$$

with $(\alpha, \beta) \in [0, 1]^2$ and they set $a = 2$ and $b = -3$. Thus, the parameters α and β control the extreme values of Γ , and the coordinates of

points p_1 and p_2 control the shape of the curve. The authors argue that if $x_0 < x_1$ and $x_2 < x_3$, inserting the solution of $t = x(\tau)$ (for $\tau(t)$) in y gives a mapping from two to one dimension. Therefore, one has to fine-tune the six hyperparameters $\vec{\theta} = (x_1, y_1, x_2, y_2, \alpha, \beta)$ to approach the optimal hopping rate Γ . However, the limitation of this method is the emergence of exponential scalings in the classical learning of $\vec{\theta}$ [40].

We compare GQW and SAMBA-GQW on unweighted and weighted MaxCut, as well as on unweighted MIS and LABS. The problem instances, described in Sec. 5.2, have even sizes $n \in [10, 20]$. As in the previous section, we proceed by Hamiltonian simulation. The classical optimization of Bézier parameters $\vec{\theta}$ is done with 100 iterations of Nelder-Mead method with bounds $\vec{\theta} \in [0, 1]^6$. We run the GQW evolution for the same evolution time T as SAMBA-GQW, and we select $q = n^2$ for the sampling protocol.

The results are presented in Fig. 29. For all considered problems, both methods achieve very high approximation ratios, with GQW consistently attaining slightly better values. Regarding the probability of measuring optimal solutions, the variational GQW also yields higher probabilities, although the differences remain relatively small. This advantage can be attributed to the use of a classical optimizer in GQW to determine the hopping-rate schedule that steers the evolution toward high-quality states, whereas SAMBA-GQW does not rely on such optimization. The probability of measuring optimal solutions is notably high for both unweighted and weighted MaxCut across all instance sizes. However, for MIS and LABS, it decreases significantly and becomes close to zero as the system size approaches $n = 20$.

The number of iterations required for GQW to reach convergence is shown in Fig. 30. Fewer than 100 iterations are sufficient for all considered problems. In particular, unweighted MaxCut and LABS exhibit a similar convergence behavior, requiring approximately 25 iterations, largely independent of the instance size. Weighted MaxCut and MIS require slightly more iterations to converge. Furthermore, Fig. 31 illustrates the number of quantum circuit evaluations as a function of the number of optimizer iterations for GQW. For all problems, a linear relationship is observed between the number of circuit evaluations and the

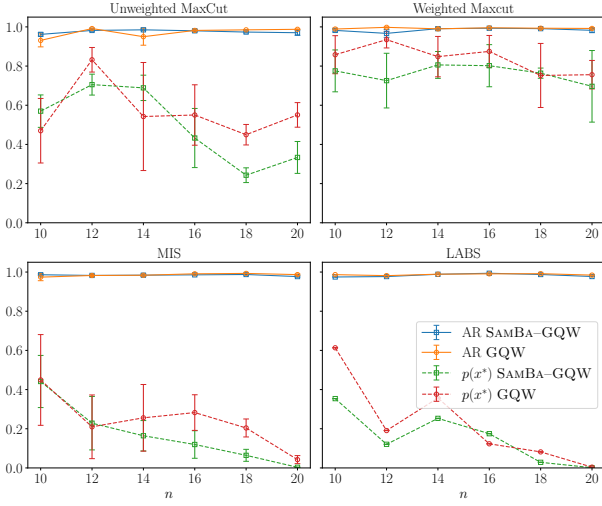


Figure 29: Approximation ratio (AR) and probability of measuring optimal solutions as a function of the number of qubits for SAMBA-GQW and GQW.

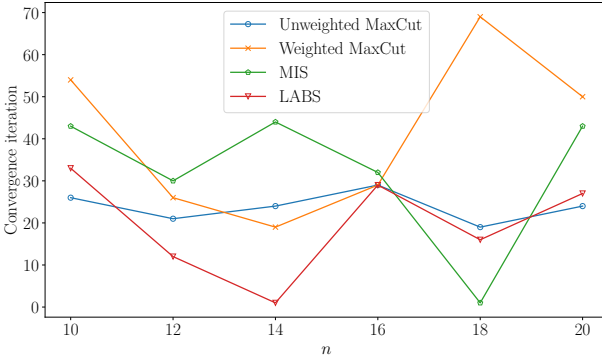


Figure 30: Number of optimization iterations required to reach convergence as a function of the number of qubits n for GQW.

number of optimization iterations. In contrast, SAMBA-GQW requires only a single quantum circuit evaluation, as it operates as a one-shot method.

In summary, although GQW achieves slightly better performance than SAMBA-GQW, the difference remains relatively small. Considering that variational GQW relies on classical optimization, which necessitates repeated quantum circuit evaluations, whereas SAMBA-GQW requires only a single evaluation, the latter offers a good trade-off between solution quality and computational cost compared to the variational approach.

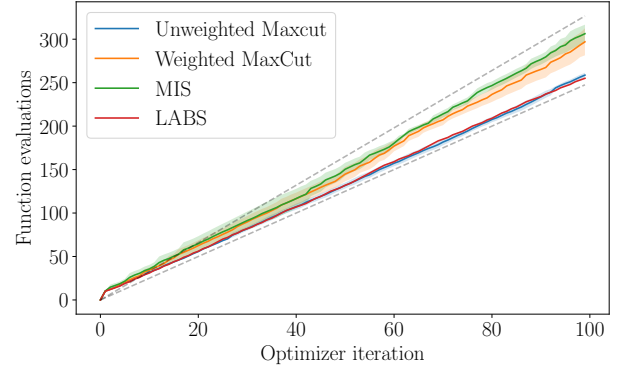


Figure 31: Number of quantum circuit evaluations as a function of the number of iterations of Nelder-Mead for GQW for $n = 20$ qubits. The dotted lines respectively denote $f(x) = 2.5x$ and $f(x) = 3.3x$.

6.2 QAOA

We present now results comparing SAMBA-GQW in its circuit form with QAOA, which is one of the most studied variational approach. This heuristic can be interpreted as a Trotterized version of quantum annealing, where the continuous-time evolution under a time-dependent Hamiltonian is replaced by a sequence of alternating unitaries generated by the problem and mixer Hamiltonians. The QAOA evolution is parametrized by a number of layers p that controls the precision of the computation. The resulting state reads:

$$|\psi_{\vec{\theta}_p}\rangle = \left(\prod_{k=1}^p e^{-i\beta_k H_M} e^{-i\gamma_k H_C} \right) |+\rangle^{\otimes n}, \quad (45)$$

and we set $\vec{\theta}_p = (\gamma_1, \beta_1, \dots, \gamma_p, \beta_p)$ and $(\gamma_k, \beta_k) \in \mathbb{R}^2$. The cost and mixer Hamiltonian are respectively defined in Eq. (2) and (4). However, one is not restricted to the use of X -mixers since a lot of different families of mixer can be of interest [8]. After the quantum evolution of Eq. (45), the expected cost value $F(\psi_{\vec{\theta}_p}) = \langle \psi_{\vec{\theta}_p} | H_C | \psi_{\vec{\theta}_p} \rangle$ is minimized classically. Then, this variational process is iterated with the updated $\vec{\theta}_p$ as long as the expectation value is above a given threshold. In the context of minimization problems, the performance of QAOA are usually measured with the approximation ratio:

$$r = \frac{C_{\min}}{F(\psi_{\vec{\theta}_p})}. \quad (46)$$

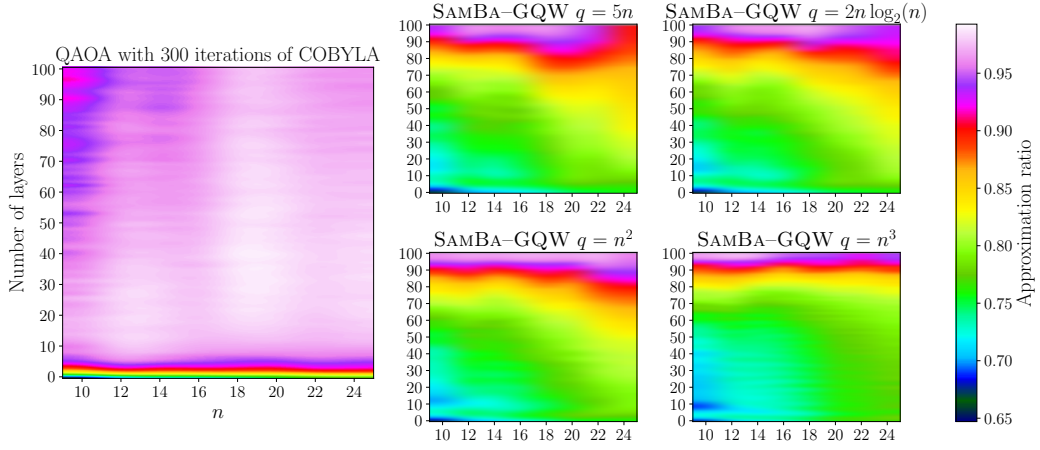


Figure 32: Approximation ratio obtained with QAOA (left) and SAMBA-GQW (right) for unweighted MaxCut on random Erdős-Rényi graphs as a function of the number of qubits number of layers of the circuits.

Since we aim at minimizing a cost function with negative values, we slightly rescale this ratio as:

$$\tilde{r} = \frac{C_{\max} - F(\psi_{\tilde{\theta}_p})}{C_{\max} - C_{\min}}. \quad (47)$$

To go further, a detailed review of QAOA and its variant can be found in Ref. [76].

We compare QAOA with SAMBA-GQW on unweighted and weighted MaxCut, as well as on unweighted MIS and LABS. The problem instances, described in Sec. 5.2, have even sizes $n \in [10, 24]$. QAOA is optimized using 300 COBYLA iterations, and the maximum number of layers is fixed to $p = 100$ for both methods⁷. Different values of the sampling parameter $q \in \{5n, 2n \log_2(n), n^2, n^3\}$ are explored for SAMBA-GQW.

The INTERP initialization strategy of Zhou et al. [7] is used for the MaxCut and MIS problems, treating optimal parameters as a smooth schedule in the layer index and initializing deeper circuits via interpolation of previously optimized angles. We use a deterministic initialization for $p = 1$, setting $\beta_1 = \pi/8$ and $\gamma_1 = 0.4/w_{\text{scale}}$, where w_{scale} is the maximum absolute adjacency matrix entry.

For the LABS problem, parameters are initialized using the dataset [77] from Ref. [15],

⁷Here, p denotes the number of circuit layers, as in standard QAOA, whereas in previous sections p_* referred to the uniform discretization parameter of SAMBA-GQW. The latter is still Trotterized as described in Sec. 3.3, with the maximum number of layers fixed to 100. Thus, QAOA and SAMBA-GQW are evaluated using circuits with the same number of layers, ensuring a fair comparison.

which provides near-optimal angles for fixed circuit depths. When the values for a given depth p are unavailable, the initialization is obtained by interpolating the parameters from depth $p - 1$.

6.2.1 MaxCut

The MaxCut cost Hamiltonian reads:

$$H_C = -\frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{ij} (\mathbb{I} - \sigma_i^z \sigma_j^z), \quad (48)$$

where $w_{ij} = 1$ for unweighted instances. The results for the unweighted and weighted MaxCut problems are summarized in Figs. 32 and 33. Overall, QAOA achieves high approximation ratios across all system sizes, with performance improving as the number of layers increases and rapidly saturating at shallow depths ($p < 10$) for both unweighted and weighted instances. For the unweighted MaxCut instances, SAMBA-GQW attains approximation ratios close to those of QAOA only at significantly larger depths ($p > 90$), with similar behavior observed across all sampling values. In contrast, for the weighted instances, increasing the sampling parameter q leads to a decrease in approximation ratios. This trend suggests that larger sampling values require substantially deeper circuits, likely well beyond the considered limit of $p = 100$ to achieve comparable performance.

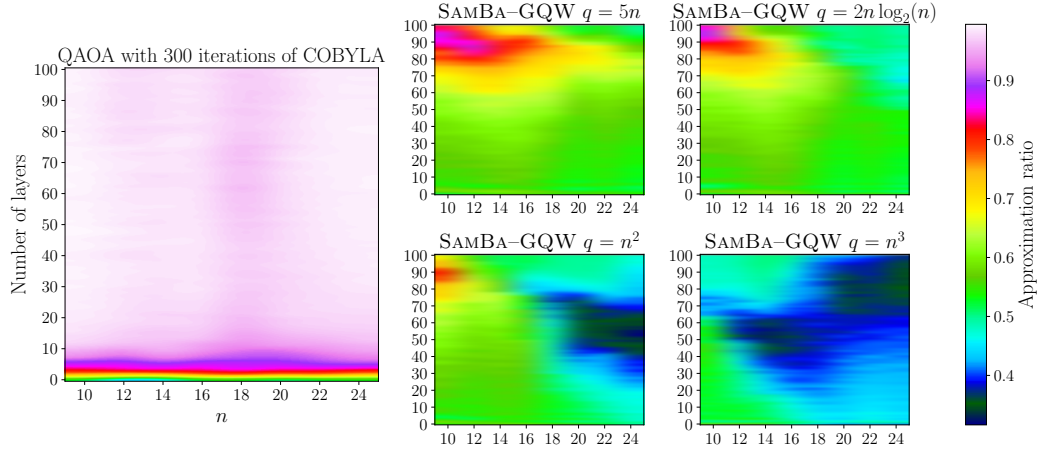


Figure 33: Approximation ratio obtained with QAOA (left) and SAMBA-GQW (right) for weighted MaxCut on random Erdős-Rényi graphs as a function of the number of qubits number of layers of the circuits.

6.2.2 Maximum independent set

The unweighted MIS cost Hamiltonian is defined as:

$$H_C = \frac{1}{2} \sum_{i \in \mathcal{V}} \sigma_i^z - \frac{\lambda}{4} \sum_{(i,j) \in \mathcal{E}} (\sigma_i^z + \sigma_j^z) + \frac{\lambda}{4} \sum_{(i,j) \in \mathcal{E}} \sigma_i^z \sigma_j^z. \quad (49)$$

The results for the unweighted MIS problem are summarized in Fig. 34. QAOA achieves high approximation ratios at shallow circuit depths ($p < 10$), with some variations across system sizes. At larger depths ($p > 20$), the approximation ratio decreases, indicating difficulties for the optimizer in effectively tuning deeper circuits. In contrast, SAMBA-GQW exhibits distinct behaviors depending on the sampling parameter q . Sampling values $q \in \{5n, 2n \log_2(n), n^2\}$ yield competitive approximation ratios that approach those of QAOA at sufficiently large depths ($p > 90$). However, further increasing the sampling value leads to a noticeable degradation in performance as the instance size grows.

6.2.3 LABS

The LABS cost Hamiltonian reads [15]:

$$H_C = 2 \sum_{i=1}^{n-3} \sigma_i^z \sum_{t=1}^{\lfloor \frac{n-i-1}{2} \rfloor} \sum_{k=t+1}^{n-i-t} \sigma_{i+t}^z \sigma_{i+k}^z \sigma_{i+k+t}^z + \sum_{i=1}^{n-2} \sigma_i^z \sum_{k=1}^{\lfloor \frac{n-i}{2} \rfloor} \sigma_{i+2k}^z. \quad (50)$$

The results for the LABS problem are summarized in Fig. 35. The LABS problem differs from MaxCut and MIS in that it possesses a large number of high-quality solutions, making it difficult to distinguish the optimal solution from the rest. As a result, the approximation ratio is already very high at $p = 0$, when the circuit consists only of a layer of Hadamard gates. QAOA achieves high approximation ratios across all system sizes, with performance steadily improving as the number of layers increases. In comparison, SAMBA-GQW exhibits a strong dependence on the sampling parameter q . Sampling values $q \in \{5n, 2n \log_2(n), n^2\}$ yield competitive results, approaching QAOA performance at large depths. Moreover, we notice that QAOA performs substantially better at shallow depths for $14 \leq n \leq 20$, whereas for larger system sizes ($n \gtrsim 20$), SAMBA-GQW achieves similar performance with fewer layers. This trend may also be explained by the growing number of high-quality solutions as n increases, which naturally raises the approximation ratio. Furthermore, increasing the sampling to $q = n^3$ leads to a noticeable degradation in approximation ratios, particularly for intermediate depths.

6.3 Overall comparison

We illustrate in Fig. 36 the best approximation ratios obtained over quantum circuits with number of layers $p \in [1, 100]$ for QAOA and SAMBA-GQW, as a function of the number of qubits. For unweighted MaxCut, QAOA consistently achieves slightly higher approximation

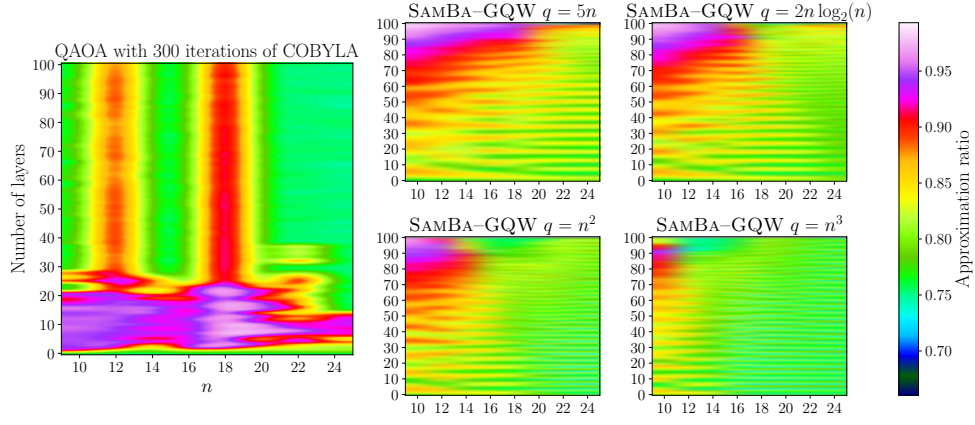


Figure 34: Approximation ratio obtained with QAOA (left) and SAMBA-GQW (right) for unweighted MIS on random Erdős-Rényi graphs as a function of the number of qubits number of layers of the circuits.

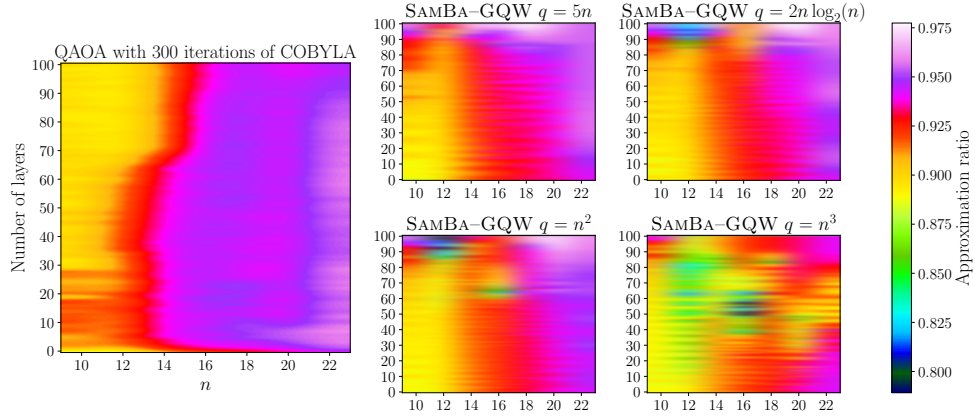


Figure 35: Approximation ratio obtained with QAOA (left) and SAMBA-GQW (right) for LABS as a function of the number of qubits and the number of layers of the circuits.

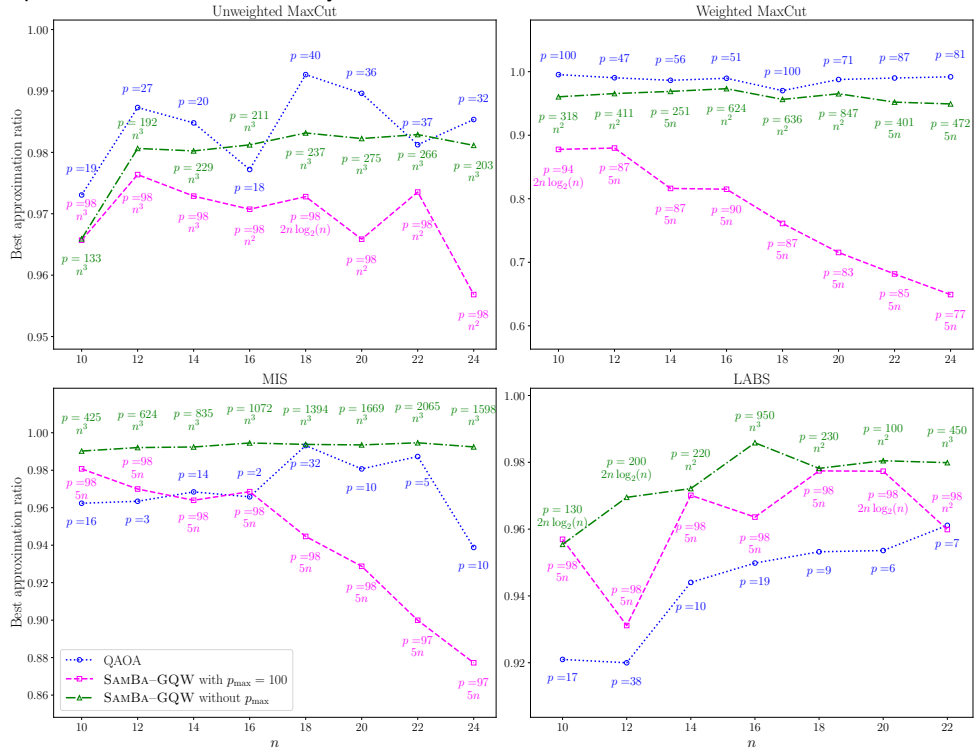


Figure 36: Best approximation ratio obtained by QAOA and SAMBA-GQW as a function of the number of qubits. Results for SAMBA-GQW are shown both with a restricted number of layers ($p_{\max} = 100$) and with an unrestricted layer budget. We indicate the number of layers p at which the highest approximation is reached, we also display the sampling parameter q that led to this performance for SAMBA-GQW.

ratios across all n . This trend becomes more pronounced for weighted MaxCut, where QAOA maintains near-optimal ratios even at large n , whereas SAMBA-GQW exhibits a marked degradation as n increases. We also observe that the best approximation ratios for QAOA are typically obtained at relatively large depths ($p \gtrsim 70$). However, as shown in Fig. 33, high-quality solutions are already reached at shallow depths ($p \approx 10$). Moreover, in contrast with the unweighted case where the optimal sampling parameter is most often $q = n^2$ or $q = n^3$, for weighted MaxCut the best performance is consistently obtained with $q = 5n$, except for $n = 10$. A similar gap is observed for MIS, where QAOA reaches higher-quality solutions at moderate depths, while SAMBA-GQW again deteriorates significantly despite using maximal circuit depths ($p = 98$). In contrast, for LABS, SAMBA-GQW behaves better than QAOA for $n < 22$, albeit at the cost of much deeper circuits. Overall, fixing the maximal number of layers to $p_{\max} = 100$ leads to QAOA performing generally better than SAMBA-GQW, achieving higher approximation ratios with shallower circuits. It is important to note, however, that QAOA relies on a classical optimization routine to determine its parameters, which contributes to the quality of the solutions, whereas SAMBA-GQW only relies on an efficient sampling protocol on the classical side. These results highlight that SAMBA-GQW is less suited to NISQ settings, as it requires deeper circuits. Finally, Fig. 36, also depicts the best approximation ratio achieved by SAMBA-GQW, with unrestricted layer budget. We see that beyond the NISQ setting, SAMBA-GQW is well in par with QAOA.

7 Conclusion

In this work, we propose a new quantum algorithm for approximating the solution of polynomial combinatorial optimization problems without using classical optimizers: SAMBA-GQW. The algorithm is based on a classical offline sampler and a quantum walk. The quantum walk is guided via a time-dependent hopping rate, which is determined a priori by the sampler. We present numerical results on several hard combinatorial problems, showcasing the high performance, consistency, and simplicity of SAMBA-GQW. Our

algorithm performs also very favorably with respect to the variational guided quantum walk, which requires a classical optimizer to determine the time-dependent hopping rate. Compared to QAOA, SAMBA-GQW behaves very similarly for approximation ratios, as long as its layer count is unrestricted (beyond NISQ regime), and less so, when forced to use shallow circuits. The latter is to be expected, since QAOA runs a classical optimizer, while SAMBA-GQW does not.

SAMBA-GQW strong performance makes it a promising heuristic candidate for solving large-scale combinatorial problems as quantum hardware continues to improve. Our approach is not intended to provide a polynomial-time solution to NP-hard problems; instead, numerical evidence suggests that polynomial resources may be sufficient to effectively guide the search toward high-quality states.

Data availability

Hamiltonian simulations were carried out with Dynamiqs [78] and quantum circuits simulations with Qiskit [79]. Quantum circuit diagrams were generated with quantikz package [80], and graphs with NetworkX [81]. The code and examples of use are available at [82].

Author contributions

U. Nzongani conceived the presented idea and carried out the simulations. D. Laplace Mermoud developed the quantum circuit discretization. U. Nzongani and D. Laplace Mermoud wrote the manuscript with support from A. Simonetto. A. Simonetto supervised the project. G. Di Molfetta has participated to several discussions, and provided his insight on the final manuscript.

Acknowledgements

This work is supported by the PEPR EPiQ ANR-22-PETQ-0007, by the ANR JCJC DisQC ANR-22-CE47-0002-01, and by the ANR project HQI-ANR-22-PNCQ-0002.

References

- [1] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärttschi, Harry Buhrman, Carleton Coffrin, Giorgio Cor-tiana, Vedran Dunjko, Daniel J Egger, Bruce G Elmegreen, et al. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, pages 1–18, 2024. DOI: [10.1038/s42254-024-00770-9](https://doi.org/10.1038/s42254-024-00770-9). URL <https://doi.org/10.1038/s42254-024-00770-9>.
- [2] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Ko-suke Mitarai, Xiao Yuan, Lukasz Cin-cio, et al. Variational quantum algo-rithms. *Nature Reviews Physics*, 3(9):625–644, 2021. URL <https://doi.org/10.1038/s42254-021-00348-9>.
- [3] Nicola Mariella and Andrea Simonetto. A quantum algorithm for the sub-graph iso-morphism problem. *ACM Transactions on Quantum Computing*, 4(2):1–34, 2023. URL <https://doi.org/10.1145/3569095>.
- [4] Dylan Laplace Mermoud, Andrea Simon-etto, and Sourour Elloumi. Variational quan-tum algorithms for permutation-based com-binatorial problems: Optimal ansatz gen-eration with applications to quadratic as-signment problems and beyond. *Quantum*, 10:1998, February 2026. ISSN 2521-327X. DOI: [10.22331/q-2026-02-09-1998](https://doi.org/10.22331/q-2026-02-09-1998). URL <https://doi.org/10.22331/q-2026-02-09-1998>.
- [5] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018. URL <https://doi.org/10.22331/q-2018-08-06-79>.
- [6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate opti-mization algorithm, 2014. URL <https://arxiv.org/abs/1411.4028>.
- [7] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum approximate optimization algo-rithm: Performance, mechanism, and imple-mentation on near-term devices. *Phys. Rev. X*, 10:021067, Jun 2020. DOI: [10.1103/PhysRevX.10.021067](https://doi.org/10.1103/PhysRevX.10.021067). URL <https://doi.org/10.1103/PhysRevX.10.021067>.
- [8] Stuart Hadfield, Zhihui Wang, Bryan O’gorman, Eleanor G Rieffel, Davide Ven-turelli, and Rupak Biswas. From the quan-tum approximate optimization algorithm to a quantum alternating operator ansatz. *Al-gorithms*, 12(2):34, 2019. URL <https://doi.org/10.3390/a12020034>.
- [9] Daniel J. Egger, Jakub Mareček, and Ste-fan Woerner. Warm-starting quantum op-timization. *Quantum*, 5:479, June 2021. ISSN 2521-327X. DOI: [10.22331/q-2021-06-17-479](https://doi.org/10.22331/q-2021-06-17-479). URL <https://doi.org/10.22331/q-2021-06-17-479>.
- [10] Yunlong Yu, Chenfeng Cao, Carter Dewey, Xiang-Bin Wang, Nic Shannon, and Robert Joynt. Quantum approximate optimiza-tion algorithm with adaptive bias fields. *Physical Review Research*, 4(2):023249, 2022. URL <https://doi.org/10.1103/PhysRevResearch.4.023249>.
- [11] Rebekah Herrman, Phillip C Lotshaw, James Ostrowski, Travis S Humble, and George Siopsis. Multi-angle quantum ap-proximate optimization algorithm. *Scientific Reports*, 12(1):6781, 2022. URL <https://doi.org/10.1038/s41598-022-10555-8>.
- [12] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Obstacles to variational quantum optimization from symmetry protection. *Phys. Rev. Lett.*, 125:260505, Dec 2020. DOI: [10.1103/PhysRevLett.125.260505](https://doi.org/10.1103/PhysRevLett.125.260505). URL <https://link.aps.org/doi/10.1103/PhysRevLett.125.260505>.
- [13] Stefan H. Sack and Maksym Serbyn. Quantum annealing initialization of the quantum approximate optimization algo-rithm. *Quantum*, 5:491, July 2021. ISSN 2521-327X. DOI: [10.22331/q-2021-07-01-491](https://doi.org/10.22331/q-2021-07-01-491). URL <https://doi.org/10.22331/q-2021-07-01-491>.
- [14] Ugo Ndongani, Dylan Laplace Mermoud, and Arthur Braida. Scaling qaoa: trans-ferring optimal adiabatic schedules from small-scale to large-scale variational cir-cuits. In *2026 International Conference on Quantum Control, Computing and Learning (qCCL)*, pages 1–6, 2026. DOI: [10.1109/qCCL70331.2026.11668574](https://doi.org/10.1109/qCCL70331.2026.11668574). URL <https://doi.org/10.1109/qCCL70331.2026.11668574>.
- [15] Ruslan Shaydulin, Changhao Li, Shouvanik

- Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, Yuri Alexeev, Joan M. Dreiling, John P. Gaebler, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Nathan Hewitt, Chandler V. Horst, Shaohan Hu, Jacob Johansen, Mitchell Matheny, Tanner Mengle, Michael Mills, Steven A. Moses, Brian Neyenhuis, Peter Siegfried, Romina Yalovetzky, and Marco Pistoia. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *Science Advances*, 10(22):eadm6761, 2024. DOI: [10.1126/sciadv.adm6761](https://doi.org/10.1126/sciadv.adm6761). URL <https://www.science.org/doi/abs/10.1126/sciadv.adm6761>.
- [16] Sami Boulebnane and Ashley Montanaro. Solving boolean satisfiability problems with the quantum approximate optimization algorithm. *PRX Quantum*, 5(3):030348, September 2024. ISSN 2691-3399. DOI: [10.1103/PRXQuantum.5.030348](https://doi.org/10.1103/PRXQuantum.5.030348). URL <https://doi.org/10.1103/PRXQuantum.5.030348>.
- [17] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J Coles, Lukasz Cincio, Jarrod R McClean, Zoë Holmes, and Marco Cerezo. Barren plateaus in variational quantum computing. *Nature Reviews Physics*, pages 1–16, 2025. URL <https://doi.org/10.1038/s42254-025-00813-9>.
- [18] Francesco De Nicola, Linda Sansoni, Andrea Crespi, Roberta Ramponi, Roberto Oselame, Vittorio Giovannetti, Rosario Fazio, Paolo Mataloni, and Fabio Sciarrino. Quantum simulation of bosonic-fermionic non-interacting particles in disordered systems via a quantum walk. *Phys. Rev. A*, 89:032322, Mar 2014. DOI: [10.1103/PhysRevA.89.032322](https://doi.org/10.1103/PhysRevA.89.032322). URL <https://link.aps.org/doi/10.1103/PhysRevA.89.032322>.
- [19] Giuseppe Di Molfetta, Marc Brachet, and Fabrice Debbasch. Quantum walks in artificial electric and gravitational fields. *Physica A: Statistical Mechanics and its Applications*, 397:157–168, 2014. ISSN 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2013.11.036>. URL <https://www.sciencedirect.com/science/article/pii/S0378437113011059>.
- [20] Giuseppe Di Molfetta, M. Brachet, and Fabrice Debbasch. Quantum walks as massless dirac fermions in curved space-time. *Phys. Rev. A*, 88:042301, Oct 2013. DOI: [10.1103/PhysRevA.88.042301](https://doi.org/10.1103/PhysRevA.88.042301). URL <https://link.aps.org/doi/10.1103/PhysRevA.88.042301>.
- [21] Julien Zylberman, Giuseppe Di Molfetta, Marc Brachet, Nuno F. Loureiro, and Fabrice Debbasch. Quantum simulations of hydrodynamics via the madelung transformation. *Phys. Rev. A*, 106:032408, Sep 2022. DOI: [10.1103/PhysRevA.106.032408](https://doi.org/10.1103/PhysRevA.106.032408). URL <https://link.aps.org/doi/10.1103/PhysRevA.106.032408>.
- [22] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 59–68, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136749. DOI: [10.1145/780542.780552](https://doi.org/10.1145/780542.780552). URL <https://doi.org/10.1145/780542.780552>.
- [23] Andrew M. Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Phys. Rev. A*, 70:022314, Aug 2004. DOI: [10.1103/PhysRevA.70.022314](https://doi.org/10.1103/PhysRevA.70.022314). URL <https://link.aps.org/doi/10.1103/PhysRevA.70.022314>.
- [24] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. DOI: [10.1137/S0097539705447311](https://doi.org/10.1137/S0097539705447311). URL <https://doi.org/10.1137/S0097539705447311>.
- [25] S. Marsh and J. B. Wang. Deterministic spatial search using alternating quantum walks. *Phys. Rev. A*, 104:022216, Aug 2021. DOI: [10.1103/PhysRevA.104.022216](https://doi.org/10.1103/PhysRevA.104.022216). URL <https://link.aps.org/doi/10.1103/PhysRevA.104.022216>.
- [26] Mathieu Roget and Giuseppe Di Molfetta. A quantum walk-based scheme for distributed searching on arbitrary graphs. In Mamata Dalui, Sukanta Das, and Enrico Formenti, editors, *Cellular Automata Technology*, pages 72–83, Cham, 2024. Springer.

- Nature Switzerland. ISBN 978-3-031-56943-2. URL https://doi.org/10.1007/978-3-031-56943-2_6.
- [27] Alvin Gonzales, Rebekah Herrman, Colin Campbell, Igor Gaidai, Ji Liu, Teague Tomesh, and Zain H Saleem. Efficient sparse state preparation via quantum walks. *npj Quantum Information*, 11(1):143, 2025. URL <https://doi.org/10.1038/s41534-025-01093-y>.
- [28] Neil B. Lovett, Sally Cooper, Matthew Everitt, Matthew Trevers, and Viv Kendon. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A*, 81:042330, Apr 2010. DOI: 10.1103/PhysRevA.81.042330. URL <https://link.aps.org/doi/10.1103/PhysRevA.81.042330>.
- [29] Andrew M. Childs. Universal computation by quantum walk. *Phys. Rev. Lett.*, 102:180501, May 2009. DOI: 10.1103/PhysRevLett.102.180501. URL <https://link.aps.org/doi/10.1103/PhysRevLett.102.180501>.
- [30] Y. Aharonov, L. Davidovich, and N. Zagury. Quantum random walks. *Phys. Rev. A*, 48:1687–1690, Aug 1993. DOI: 10.1103/PhysRevA.48.1687. URL <https://link.aps.org/doi/10.1103/PhysRevA.48.1687>.
- [31] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Phys. Rev. A*, 58:915–928, Aug 1998. DOI: 10.1103/PhysRevA.58.915. URL <https://link.aps.org/doi/10.1103/PhysRevA.58.915>.
- [32] Jessica Lemieux, Bettina Heim, David Poulin, Krysta Svore, and Matthias Troyer. Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm. *Quantum*, 4:287, June 2020. ISSN 2521-327X. DOI: 10.22331/q-2020-06-29-287. URL <https://doi.org/10.22331/q-2020-06-29-287>.
- [33] Roberto Campos, Pablo AM Casares, and MA Martin-Delgado. Quantum metropolis solver: a quantum walks approach to optimization problems. *Quantum Machine Intelligence*, 5(2):28, 2023. URL <https://doi.org/10.1007/s42484-023-00119-y>.
- [34] Adam Callison, Nicholas Chancellor, Florian Mintert, and Viv Kendon. Finding spin glass ground states using quantum walks. *New Journal of Physics*, 21(12):123022, dec 2019. DOI: 10.1088/1367-2630/ab5ca2. URL <https://doi.org/10.1088/1367-2630/ab5ca2>.
- [35] S. Marsh and J. B. Wang. Combinatorial optimization via highly efficient quantum walks. *Phys. Rev. Res.*, 2:023302, Jun 2020. DOI: 10.1103/PhysRevResearch.2.023302. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.2.023302>.
- [36] T. Bennett, E. Matwiejew, S. Marsh, and J. B. Wang. Quantum walk-based vehicle routing optimisation. *Frontiers in Physics*, Volume 9 - 2021, 2021. ISSN 2296-424X. DOI: 10.3389/fphy.2021.730856. URL <https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2021.730856>.
- [37] N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang. Quantum walk-based portfolio optimisation. *Quantum*, 5:513, July 2021. ISSN 2521-327X. DOI: 10.22331/q-2021-07-28-513. URL <https://doi.org/10.22331/q-2021-07-28-513>.
- [38] Adam Callison, Max Festenstein, Jie Chen, Laurentiu Nita, Viv Kendon, and Nicholas Chancellor. Energetic perspective on rapid quenches in quantum annealing. *PRX Quantum*, 2:010338, Mar 2021. DOI: 10.1103/PRXQuantum.2.010338. URL <https://link.aps.org/doi/10.1103/PRXQuantum.2.010338>.
- [39] Robert J. Banks, Ehsan Haque, Farah Nazef, Fatima Fethallah, Fatima Ruqaya, Hamza Ahsan, Het Vora, Hibah Tahir, Ibrahim Ahmad, Isaac Hewins, Ishaq Shah, Krish Baranwal, Mannan Arora, Mateen Asad, Mubasshirah Khan, Nabian Hasan, Nuh Azad, Salgai Fedaiee, Shakeel Majeed, Shayam Bhuyan, Tasfia Tarannum, Yahya Ali, Dan E. Browne, and P. A. Warburton. Continuous-time quantum walks for MAX-CUT are hot. *Quantum*, 8:1254, February 2024. ISSN 2521-327X. DOI: 10.22331/q-2024-02-13-1254. URL <https://doi.org/10.22331/q-2024-02-13-1254>.
- [40] Sebastian Schulz, Dennis Willsch, and Kristel Michielsen. Guided quantum walk. *Phys. Rev. Res.*, 6:013312, Mar 2024. DOI: 10.1103/PhysRevResearch.6.013312. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.6.013312>.

- [41] Robert J. Banks, Georgios S. Raftis, Dan E. Browne, and P. A. Warburton. Continuous-time quantum optimization without the adiabatic principle. *Phys. Rev. A*, 111:032414, Mar 2025. DOI: [10.1103/PhysRevA.111.032414](https://link.aps.org/doi/10.1103/PhysRevA.111.032414). URL <https://link.aps.org/doi/10.1103/PhysRevA.111.032414>.
- [42] Alejandro Gomez Cadavid, Archismita Dalal, Anton Simen, Enrique Solano, and Narendra N. Hegade. Bias-field digitized counterdiabatic quantum optimization. *Phys. Rev. Res.*, 7:L022010, Apr 2025. DOI: [10.1103/PhysRevResearch.7.L022010](https://link.aps.org/doi/10.1103/PhysRevResearch.7.L022010). URL <https://link.aps.org/doi/10.1103/PhysRevResearch.7.L022010>.
- [43] Bin Yan and Nikolai A Sinitsyn. Analytical solution for nonadiabatic quantum annealing to arbitrary Ising spin Hamiltonian. *Nature Communications*, 13(1):2212, 2022. URL <https://doi.org/10.1038/s41467-022-29887-0>.
- [44] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000. URL <https://doi.org/10.48550/arXiv.quant-ph/0001106>.
- [45] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90:015002, Jan 2018. DOI: [10.1103/RevModPhys.90.015002](https://link.aps.org/doi/10.1103/RevModPhys.90.015002). URL <https://link.aps.org/doi/10.1103/RevModPhys.90.015002>.
- [46] David R. Burt, Carl Edward Rasmussen, and Mark van der Wilk. Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020. URL <http://jmlr.org/papers/v21/19-1015.html>.
- [47] Masuo Suzuki. Decomposition formulas of exponential operators and lie exponentials with some applications to quantum mechanics and statistical physics. *Journal of mathematical physics*, 26(4):601–612, 1985. URL <https://doi.org/10.1063/1.526596>.
- [48] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021. DOI: [10.1103/PhysRevX.11.011020](https://link.aps.org/doi/10.1103/PhysRevX.11.011020). URL <https://link.aps.org/doi/10.1103/PhysRevX.11.011020>.
- [49] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2, 2014. ISSN 2296-424X. DOI: [10.3389/fphy.2014.00005](https://doi.org/10.3389/fphy.2014.00005). URL <http://dx.doi.org/10.3389/fphy.2014.00005>.
- [50] Zoé Verchère, Sourour Elloumi, and Andrea Simonetto. Optimizing variational circuits for higher-order binary optimization. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 01, pages 19–25, 2023. DOI: [10.1109/QCE57702.2023.00011](https://doi.org/10.1109/QCE57702.2023.00011). URL <https://doi.org/10.1109/QCE57702.2023.00011>.
- [51] Julien Zylberman, Ugo Nzongani, Andrea Simonetto, and Fabrice Debbasch. Efficient quantum circuits for non-unitary and unitary diagonal operators with space-time-accuracy trade-offs. *ACM Transactions on Quantum Computing*, 6(2), April 2025. DOI: [10.1145/3718348](https://doi.org/10.1145/3718348). URL <https://doi.org/10.1145/3718348>.
- [52] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1):165–177, 1990. ISSN 0012-365X. DOI: [https://doi.org/10.1016/0012-365X\(90\)90358-O](https://doi.org/10.1016/0012-365X(90)90358-O). URL <https://www.sciencedirect.com/science/article/pii/0012365X90903580>.
- [53] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin. Quantum optimization of maximum independent set using rydberg atom arrays. *Science*, 376(6598):1209–1215, 2022. DOI: [10.1126/science.abo6587](https://doi.org/10.1126/science.abo6587). URL <https://www.science.org/doi/abs/10.1126/science.abo6587>.
- [54] Constantin Dalyac, Lucas Leclerc, Louis Vignoli, Mehdi Djellabi, Wesley da Silva Coelho, Bruno Ximenez, Alexandre Dareau, Davide Dreoni, Vincent E Elfving, Adrien Signoles, et al. Graph algorithms with neutral atom quantum processors. *The*

- European Physical Journal A*, 60(9):177, 2024. URL <https://doi.org/10.1140/epja/s10050-024-01385-5>.
- [55] Constantin Dalyac, Loïc Henriët, Emmanuel Jeandel, Wolfgang Lechner, Simon Perdrix, Marc Porcheron, and Margarita Veshcheze-rova. Qualifying quantum approaches for hard industrial optimization problems. a case study in the field of smart-charging of electric vehicles. *EPJ Quantum Technology*, 8(1):12, 2021. URL <https://doi.org/10.1140/epjqt/s40507-021-00100-3>.
- [56] Lucas Leclerc, Constantin Dalyac, Pascale Bendotti, Rodolphe Griset, Joseph Mikael, and Loic Henriët. Implementing transferable annealing protocols for combinatorial optimization on neutral-atom quantum processors: A case study on smart charging of electric vehicles. *Phys. Rev. A*, 111:032611, Mar 2025. DOI: [10.1103/PhysRevA.111.032611](https://doi.org/10.1103/PhysRevA.111.032611). URL <https://link.aps.org/doi/10.1103/PhysRevA.111.032611>.
- [57] Ruben S. Andrist, Martin J. A. Schuetz, Pierre Minssen, Romina Yalovetzky, Shouvanik Chakrabarti, Dylan Herman, Niraj Kumar, Grant Salton, Ruslan Shaydulín, Yue Sun, Marco Pistoia, and Helmut G. Katzgraber. Hardness of the maximum-independent-set problem on unit-disk graphs and prospects for quantum speedups. *Phys. Rev. Res.*, 5:043277, Dec 2023. DOI: [10.1103/PhysRevResearch.5.043277](https://doi.org/10.1103/PhysRevResearch.5.043277). URL <https://link.aps.org/doi/10.1103/PhysRevResearch.5.043277>.
- [58] Pierre Cazals, Aymeric François, Loïc Henriët, Lucas Leclerc, Malory Marin, Yassine Naghmouchi, Wesley da Silva Coelho, Florian Sikora, Vittorio Vitale, Rémi Watrigant, Monique Witt Garzillo, and Constantin Dalyac. Identifying hard native instances for the maximum-independent-set problem on neutral-atom quantum processors. *Phys. Rev. Appl.*, 25:034085, Mar 2026. DOI: [10.1103/8gfv-lllf](https://doi.org/10.1103/8gfv-lllf). URL <https://link.aps.org/doi/10.1103/8gfv-lllf>.
- [59] Victor A. Penas, Marcelo Losada, and Pedro W. Lamberti. Quantum imaginary time evolution and the unit-disk maximum independent set problem. *Phys. Rev. A*, 111:042403, Apr 2025. DOI: [10.1103/PhysRevA.111.042403](https://doi.org/10.1103/PhysRevA.111.042403). URL <https://link.aps.org/doi/10.1103/PhysRevA.111.042403>.
- [60] Zhihui Wang, Nicholas C. Rubin, Jason M. Dominy, and Eleanor G. Rieffel. xy mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A*, 101:012320, Jan 2020. DOI: [10.1103/PhysRevA.101.012320](https://doi.org/10.1103/PhysRevA.101.012320). URL <https://link.aps.org/doi/10.1103/PhysRevA.101.012320>.
- [61] Andreas Bärttschi and Stephan Eidenbenz. Deterministic preparation of dicke states. In Leszek Antoni Gasieniec, Jesper Jansson, and Christos Levcopoulos, editors, *Fundamentals of Computation Theory*, pages 126–139, Cham, 2019. Springer International Publishing. ISBN 978-3-030-25027-0. URL https://doi.org/10.1007/978-3-030-25027-0_9.
- [62] Tom Packebusch and Stephan Mertens. Low autocorrelation binary sequences. *Journal of Physics A: Mathematical and Theoretical*, 49(16):165001, mar 2016. DOI: [10.1088/1751-8113/49/16/165001](https://doi.org/10.1088/1751-8113/49/16/165001). URL <https://doi.org/10.1088/1751-8113/49/16/165001>.
- [63] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery. ISBN 9781450374644. DOI: [10.1145/800157.805047](https://doi.org/10.1145/800157.805047). URL <https://doi.org/10.1145/800157.805047>.
- [64] Elijah Pelofske, Andreas Bärttschi, John Golden, and Stephan Eidenbenz. High-round qaoa for max k -sat on trapped ion nisq devices. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 01, pages 506–517, 2023. DOI: [10.1109/QCE57702.2023.00064](https://doi.org/10.1109/QCE57702.2023.00064). URL <https://doi.org/10.1109/QCE57702.2023.00064>.
- [65] Yunlong Yu, Chenfeng Cao, Xiang-Bin Wang, Nic Shannon, and Robert Joynt. Solution of sat problems with the adaptive-bias quantum approximate optimization algorithm. *Phys. Rev. Res.*, 5:023147, Jun 2023. DOI: [10.1103/PhysRevResearch.5.023147](https://doi.org/10.1103/PhysRevResearch.5.023147).

- URL <https://link.aps.org/doi/10.1103/PhysRevResearch.5.023147>.
- [66] Rajesh Matai, Surya Singh, and Murari Lal Mittal. Traveling salesman problem: an overview of applications, formulations, and solution approaches. In Donald Davendra, editor, *Traveling Salesman Problem - Theory and Applications*, chapter 1. IntechOpen, London, 2010. DOI: [10.5772/12909](https://doi.org/10.5772/12909). URL <https://doi.org/10.5772/12909>.
- [67] Adam Glos, Aleksandra Krawiec, and Zoltán Zimborás. Space-efficient binary optimization for variational quantum computing. *npj Quantum Information*, 8(1):39, 2022. URL <https://doi.org/10.1038/s41534-022-00546-y>.
- [68] Paul Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17, 1960.
- [69] Thorsten Koch, David E Bernal Neira, Ying Chen, Giorgio Cortiana, Daniel J Egger, Raoul Heese, Narendra N Hegade, Alejandro Gomez Cadavid, Rhea Huang, Toshinari Itoko, et al. The quantum optimization benchmarking library. *Nature Computational Science*, 6(6):653–671, 2026. URL <https://doi.org/10.1038/s43588-026-00991-1>.
- [70] Quantum Optimization Benchmarking Library (QOBLIB), 2026. URL <https://doi.org/10.5281/zenodo.18962381>. Zenodo.
- [71] Przemysław Juszczuk, Ignacy Kaliszewski, Janusz Miroforidis, and Dmitry Podkopaev. Mean-variance data collections for portfolio optimization problems, 2022. URL <https://doi.org/10.18150/CZYLOV>.
- [72] Earl Campbell, Ankur Khurana, and Ashley Montanaro. Applying quantum algorithms to constraint satisfaction problems. *Quantum*, 3:167, July 2019. ISSN 2521-327X. DOI: [10.22331/q-2019-07-18-167](https://doi.org/10.22331/q-2019-07-18-167). URL <https://doi.org/10.22331/q-2019-07-18-167>.
- [73] Rémi Monasson and Riccardo Zecchina. Entropy of the k-satisfiability problem. *Phys. Rev. Lett.*, 76:3881–3885, May 1996. DOI: [10.1103/PhysRevLett.76.3881](https://link.aps.org/doi/10.1103/PhysRevLett.76.3881). URL <https://link.aps.org/doi/10.1103/PhysRevLett.76.3881>.
- [74] Stephan Mertens, Marc Mézard, and Riccardo Zecchina. Threshold values of random k-sat from the cavity method. *Random Structures & Algorithms*, 28(3):340–373, 2006. DOI: <https://doi.org/10.1002/rsa.20090>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.20090>.
- [75] Amin Coja-Oghlan. The asymptotic k-sat threshold. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 804–813, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327107. DOI: [10.1145/2591796.2591822](https://doi.org/10.1145/2591796.2591822). URL <https://doi.org/10.1145/2591796.2591822>.
- [76] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. A review on quantum approximate optimization algorithm and its variants. *Physics Reports*, 1068:1–66, 2024. ISSN 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2024.03.002>. URL <https://www.sciencedirect.com/science/article/pii/S0370157324001078>. A review on Quantum Approximate Optimization Algorithm and its variants.
- [77] Ruslan Shaydulin, Changhao Li, Shouvanik Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, Yuri Alexeev, Joan M. Dreiling, John P. Gaebler, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Nathan Hewitt, Chandler V. Horst, Shaohan Hu, Jacob Johansen, Mitchell Matheny, Tanner Mingle, Michael Mills, Steven A. Moses, Brian Neyenhuis, Peter Siegfried, Romina Yalovetzky, and Marco Pistoia. Data from "evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem", July 2023. URL <https://doi.org/10.5281/zenodo.8190275>.
- [78] Pierre Guilmin, Adrien Bocquet, Élie Genois, Daniel Weiss, and Ronan Gautier. Dynamiqs: an open-source python library for gpu-accelerated and differentiable simulation of quantum systems. 2025. URL <https://github.com/dynamiqs/dynamiqs>.
- [79] Ali Javadi-Abhari, Matthew Treinish, Kevin

Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024. URL <https://doi.org/10.48550/arXiv.2405.08810>.

- [80] Alastair Kay. Tutorial on the quantikz package. *arXiv preprint arXiv:1809.03842*, 2018. URL <https://doi.org/10.48550/arXiv.1809.03842>.
- [81] Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008. URL <https://doi.org/10.25080/TCWV9851>.
- [82] Samba-gqw. <https://github.com/ugo-nzongani/SamBa-GQW>, 2025.

Appendices

A Hopping rate as a function of energy

The hopping rate Γ is initially a function of energy, which corresponds to the output of the cost function, which is the reason why we can consider “negative energies”. We make it a function of time to define a QW on the solution space, whose connectivity is defined by the structure of the mixer. Since each energy requires a specific transfer time (see Eq. (11)), the original function of energy $\Gamma(E)$ is rescaled to obtain the time-dependent hopping rate $\Gamma(t)$. We show the exact ($q = 2^n$) and approximated ($q = n^2$) original function of energy on Fig. 37 for $n = 11$ qubits. Unlike the time-dependent hopping rates displayed on Figs. 1b and 1d, it is clear that when it depends on energy, and are therefore not rescaled, the exact and approximate hopping rates are very similar.

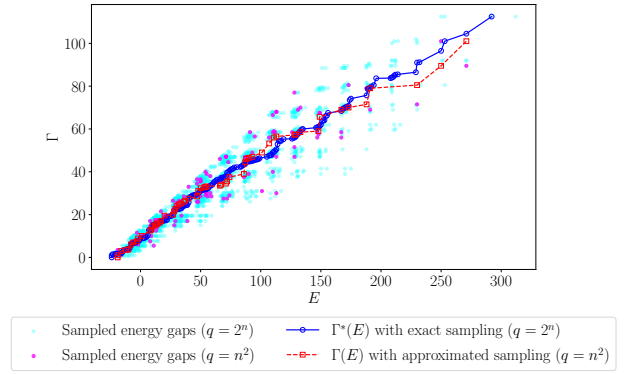


Figure 37: Hopping rate Γ as a function of energy E for $n = 11$ qubits with $\Gamma(E) = \frac{\langle \Delta^C \rangle(E)}{2}$ for UD-MIS instance.

B Uniform discretization of Γ

In our numerical simulations we split each linear sub-interval into $p_\star$ cuts, we illustrate such a discretization scheme with $p_\star = 2$ in Fig. 38.

C Sampled energy scaling with n

In this section, we show the number of different sampled energies $|\mathcal{E}|$ as a function of the number of qubits for MaxCut (Fig. 39), MIS (Fig. 40) and LABS (Fig. 41). As n increases, this quantity approaches zero for LABS, explaining the behavior

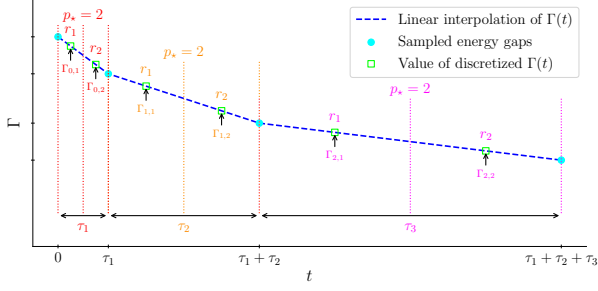


Figure 38: Uniform discretization of the hopping rate Γ .

observed in Fig. 24. In Appendix E, we show how to prevent this vanishing behavior.

D Deeper quantum circuits for SAMBA-GQW

In this section, we illustrate the effect of using higher-depth circuits for weighted MaxCut and UD-MIS on $n = 22$ qubits by extending the discretization parameter to the range $p_* \in [0, 30]$. As shown in Figs. 42 and 43, increasing the circuit depth leads to the best performance when sampling with $q = n^3$, although for the weighted MaxCut problem the results are nearly identical to those obtained with other sampling strategies. However, despite the improvement in the approximation ratio, the number of circuit layers increases significantly, making this approach inefficient. Therefore, it does not appear necessary to use large values of the discretization parameter p_* , as the gain in solution quality is negligible compared to the substantial increase in circuit depth.

E Absolute values of the energy gaps

In the sampling protocol presented in Algorithm 2, we consider only positive energy gaps. More precisely, at each iteration, a state is randomly selected and its energy gaps with all neighboring states are computed; only the positive gaps are retained, and the largest among them is stored. One may naturally ask why we do not instead consider both positive and negative gaps by taking their absolute values, since the sign of a gap depends solely on the randomly chosen reference state and does not carry intrinsic information. This choice was made to reduce the number of sampled gaps and, consequently, lower

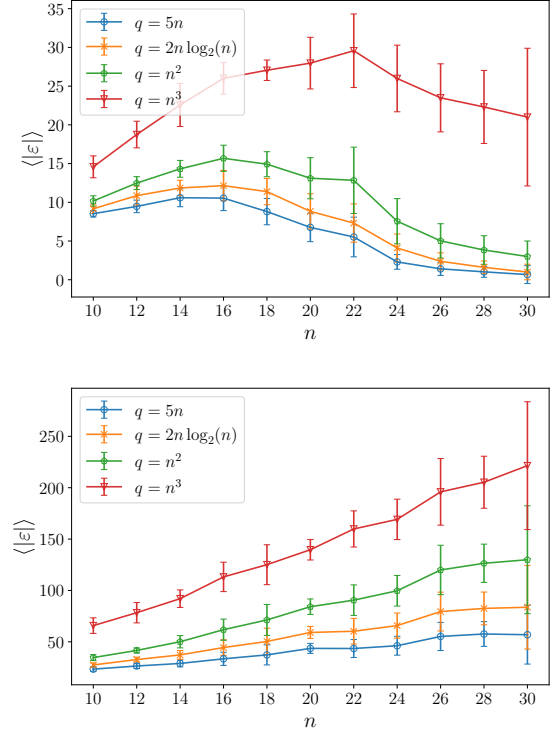


Figure 39: Number of different sampled energies for unweighted (top) and weighted (bottom) MaxCut on random Erdős-Rényi graphs for $n \in [10, 30]$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

the computational cost of the classical simulations. Numerical results indicate that, for most of the considered optimization problems, this restricted sampling strategy remains sufficient to obtain high-quality solutions. However, as shown in Figs. 39 (unweighted MaxCut) and 41, the number of sampled gaps decreases as n increases. In the worst-case scenario, this may lead to situations where no energy gaps are sampled at all, implying that no information is extracted to construct the hopping rate and, consequently, no information is available to determine the circuit angles.

In this section, we modify the sampling protocol by considering the absolute values of the energy gaps rather than restricting to positive ones, and we study the impact of this modification on the results for unweighted MaxCut and LABS. These two problems were selected because they are the only cases in our simulations where the number of sampled gaps decreases significantly with increasing problem size. With this modifi-

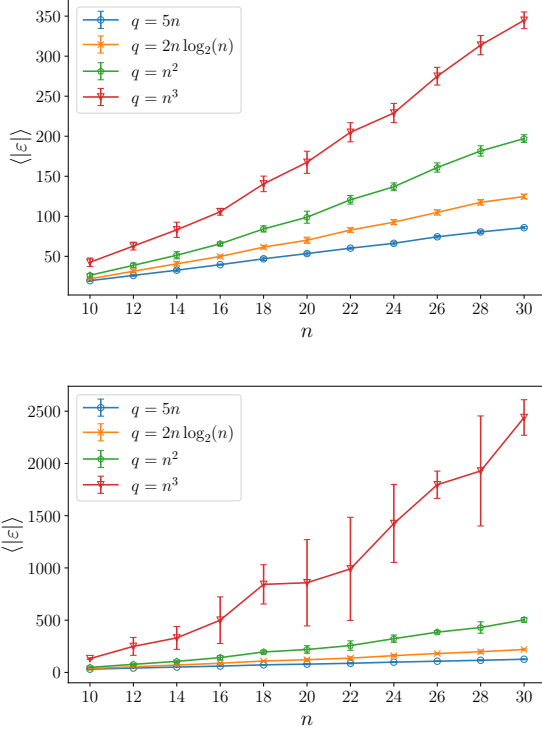


Figure 40: Number of different sampled energies for unweighted MIS on random Erdős-Rényi graphs (top) and weighted MIS on unit-disk graphs (bottom) for $n \in [10, 30]$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

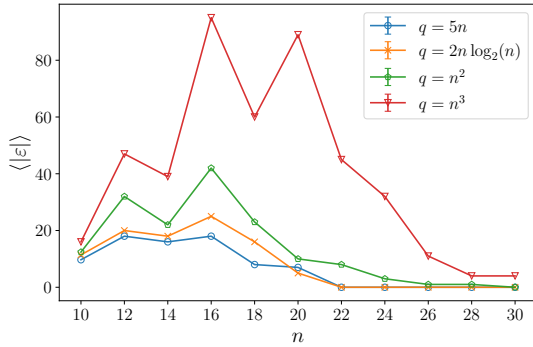


Figure 41: Number of different sampled energies for LABS for $n \in [10, 30]$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

cation, line 7 of Algorithm 2 becomes

$$\Delta \leftarrow \left\{ \frac{|\Delta_{x,y}^C|}{|\Delta_{x,y}^M|} \mid \forall y \in B(x) \right\}. \quad (51)$$

We present the results of the quantum circuit simulations for unweighted MaxCut for $n = 24$ qubits on Fig. 44. Unsurprisingly, relaxing the restriction to positive gaps results in a larger number of sampled gaps. This modification also impacts the approximation ratio, as sampling only positive gaps leads to systematically higher values, with a discrepancy of roughly 0.2 between the two methods. In addition, Fig. 45 indicates that, when gap magnitudes are considered irrespective of their sign, the number of sampled gaps continues to increase as the problem size grows. Furthermore, we illustrate the approximation ratio and number of layers in the quantum circuits as a function of n in Fig. 46. We also present the results for LABS with $n = 26$ qubits in Fig. 47. As in the case of unweighted MaxCut (see Fig. 44), including both positive and negative gaps leads to a substantially larger number of sampled gaps, increasing from zero to approximately 50 for $q \in \{5n, 2n \log_2(n)\}$. Consistent with the observations for unweighted MaxCut, considering all gaps together with their sign results in lower approximation ratios than those obtained when only positive gaps are considered. However, in contrast to unweighted MaxCut, the difference remains below 0.1. Furthermore, Fig. 48 illustrates the evolution of the number of sampled gaps as a function of the problem size. Although this quantity exhibits oscillatory behavior, alternating between increases and decreases as n grows, the magnitude of the increases appears to consistently exceed that of the decreases, resulting in an overall upward trend. Finally, Fig. 49 reports the approximation ratio and the number of layers in the quantum circuits as functions of the problem size n .

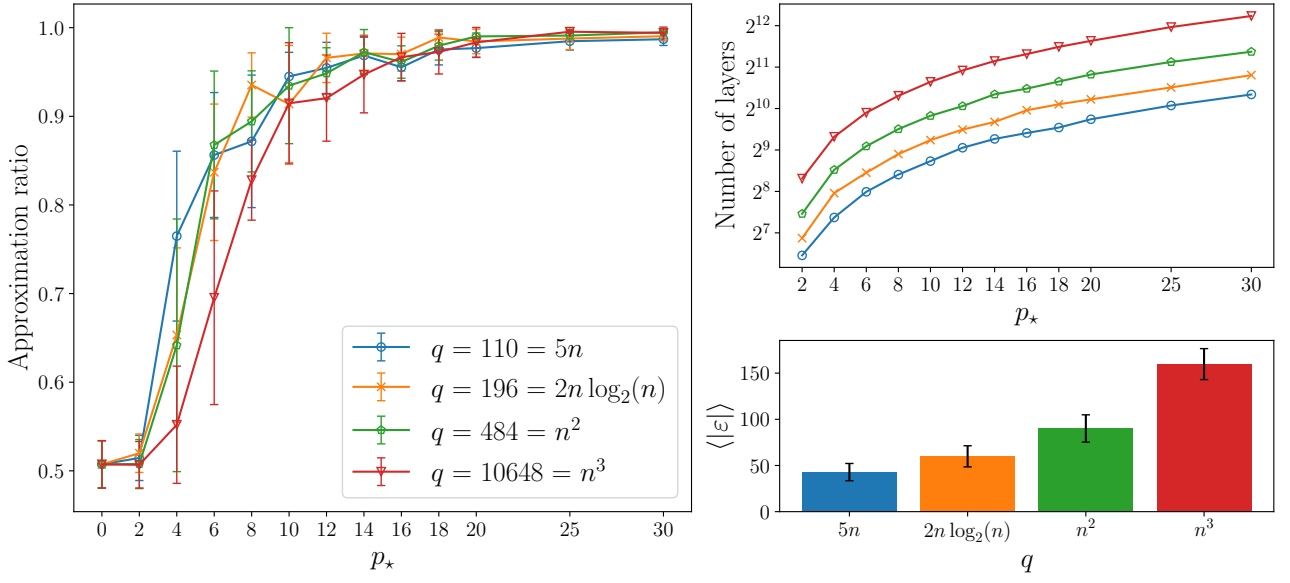


Figure 42: SAMBA-GQW quantum circuit simulations for weighted MaxCut on random Erdős-Rényi graphs for $n = 22$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The discretization parameter p_* takes values in interval $[0, 30]$.

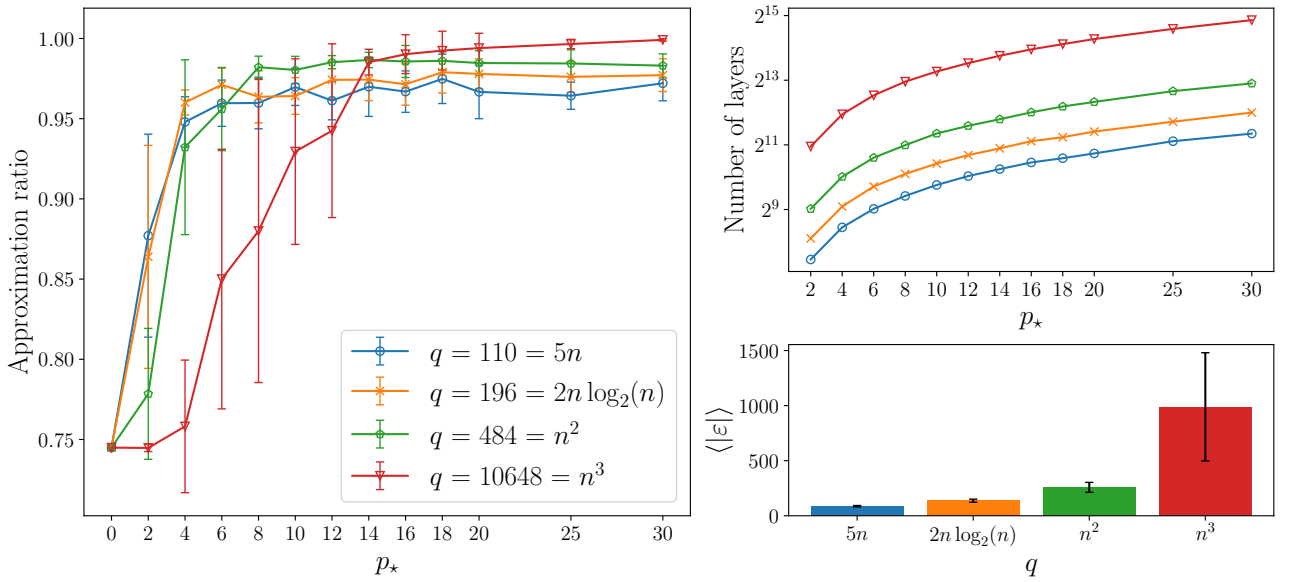


Figure 43: SAMBA-GQW quantum circuit simulations for weighted MIS on unit-disk graphs for $n = 22$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The discretization parameter p_* takes values in interval $[0, 30]$.

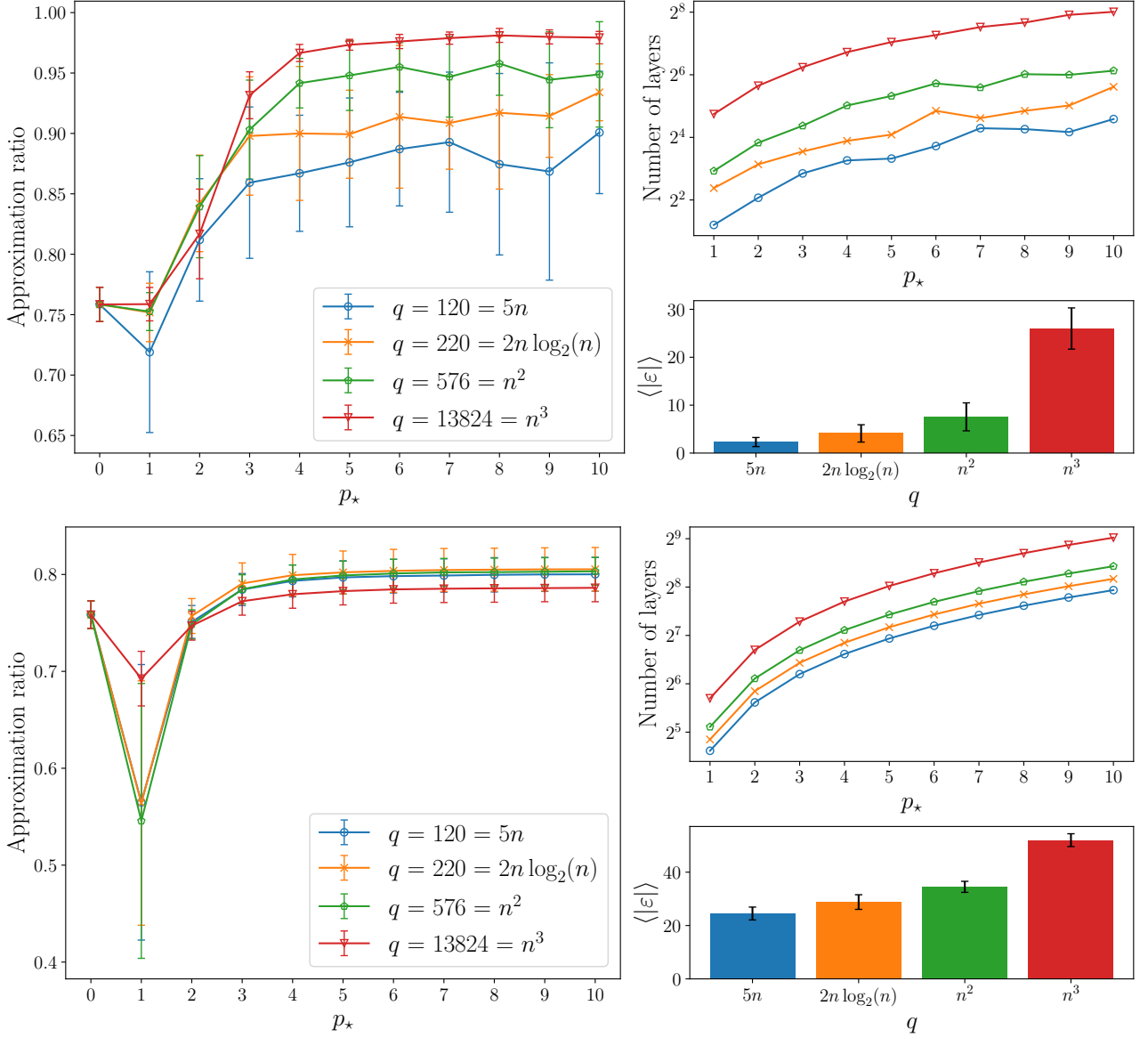


Figure 44: SAMBA-GQW quantum circuit simulations for unweighted MaxCut on random Erdős-Rényi graphs for $n = 24$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The discretization parameter p_* takes values in interval $[0, 10]$. The upper (resp. lower) figure presents the results obtained when considering only positive gaps (resp. both positive and negative gaps).

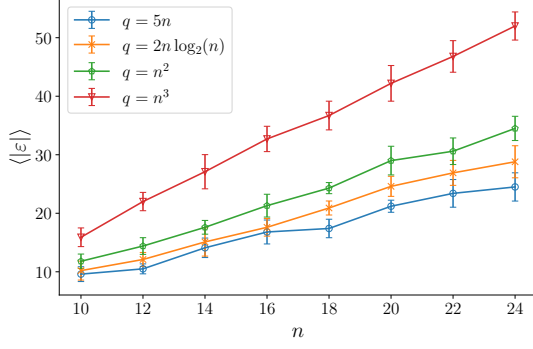


Figure 45: SAMBA-GQW quantum circuit simulations for unweighted MaxCut on random Erdős-Rényi graphs for $n \in [10, 24]$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The discretization parameter p_* takes values in interval $[0, 10]$.

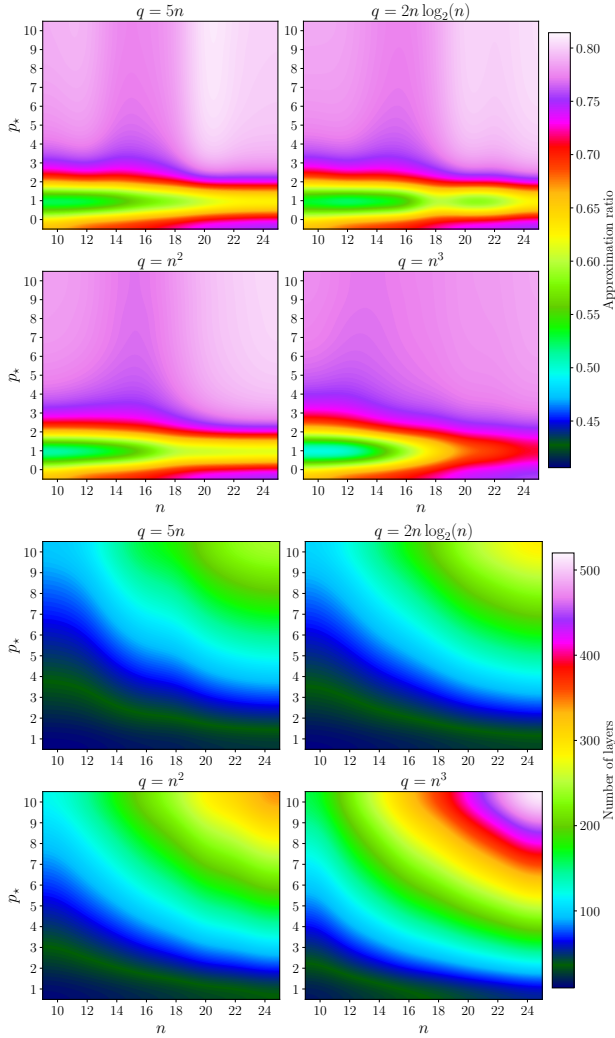


Figure 46: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 24]$ and the discretization parameter $p_* \in [1, 10]$ for unweighted MaxCut on random Erdős-Rényi graphs with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.

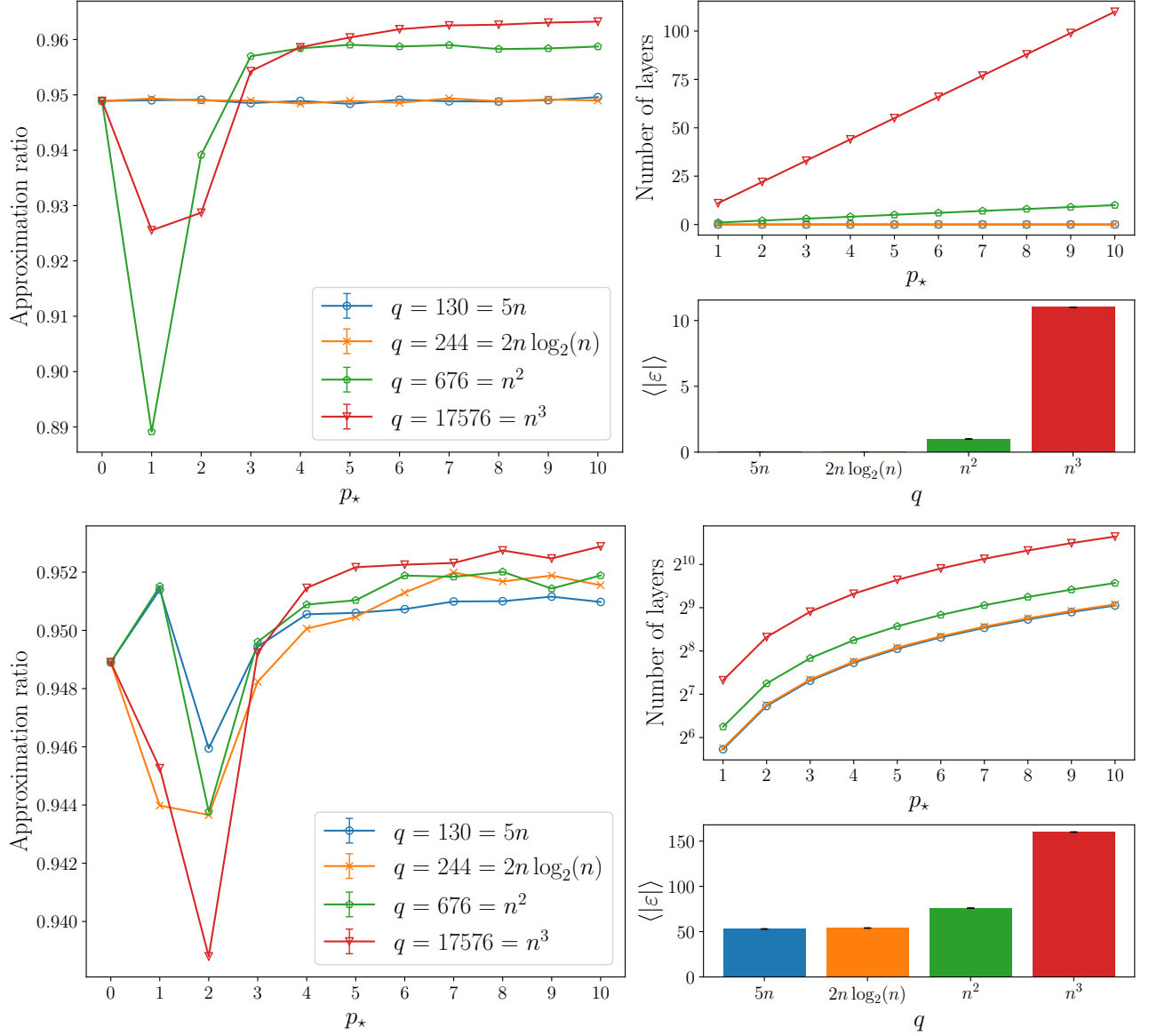


Figure 47: SAMBA-GQW quantum circuit simulations for LABS for $n = 26$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The discretization parameter p_* takes values in interval $[0, 10]$. The upper (resp. lower) figure presents the results obtained when considering only positive gaps (resp. both positive and negative gaps).

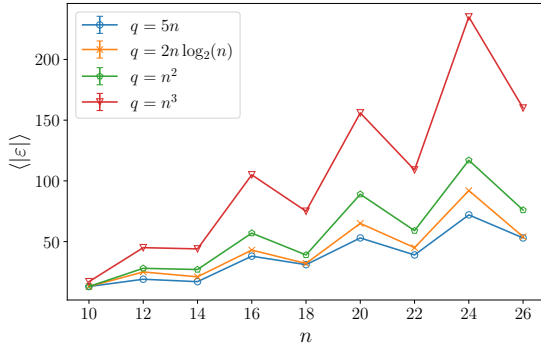


Figure 48: SAMBA-GQW quantum circuit simulations for LABS for $n \in [10, 26]$ qubits with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$. The discretization parameter $p_\star$ takes values in interval $[0, 10]$.

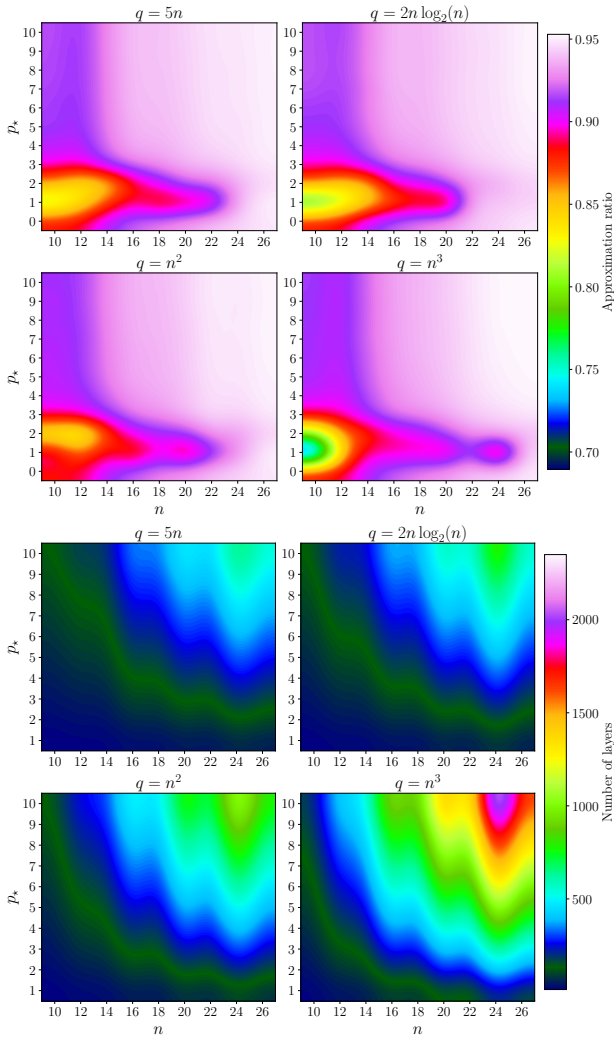


Figure 49: Approximation ratio (top) and number of layers in the circuit (bottom) as a function of the number of qubits $n \in [10, 26]$ and the discretization parameter $p_\star \in [1, 10]$ for LABS with different sampling values $q \in \{5n, 2n \log_2(n), n^2, n^3\}$.