IEEE Transactions on Quantum Engineering

Received 1 August 2024; revised 14 February 2025; accepted 19 March 2025; date of publication 28 March 2025;
date of current version 24 April 2025.

Digital Object Identifier 10.1109/TQE.2025.3555562

Mixed Grover: A Hybrid Version to Improve Grover's Algorithm for Unstructured Database Search

ROMAIN PIRON (Student Member, IEEE), MUHAMMAD IDHAM HABIBIE (Student Member, IEEE), AND CLAIRE GOURSAUD (Member, IEEE)

INSA Lyon, Inria, CITI EA 3720, Université de Lyon, 69007 Lyon, France

Corresponding Author: Romain Piron (e-mail: romain.piron@insa-lyon.fr).

This work was supported by the French National Research Agency (ANR) through the France 2030 program under Grant NF-PERSEUS: ANR-22-PEFT-0004.

ABSTRACT In this article, we propose a new strategy to exploit Grover's algorithm for unstructured search problems. We first show that running Grover's routine with a reduced number of iterations but allowing several trials presents a complexity advantage while keeping the same success probability. Then, by a theoretical analysis of the performance, we provide a generic procedure to parameterize the number of iterations k within one shot of Grover's algorithm and the maximum number of trials T, given a targeted success p and the size of the database N. At the end, we highlight that this new approach permits to reduce the computational time by at least 10% for $p \geq 0.999$ independently of the size of the database.

INDEX TERMS Complexity, grover's algorithm, hybrid quantum computing, unstructured search.

I. INTRODUCTION

Grover's algorithm stands as a cornerstone in quantum computing for solving unstructured search problems. Introduced by Grover [1], it has garnered widespread attention for its potential applications across various problem domains. The initial formulation of the algorithm proposes an iterative method that allows to find a marked element x_0 in an unsorted database with a certain success probability p. At each iteration, the algorithm calls a quantum oracle [1]. An appropriate parameterization of Grover's algorithm allows to reach a success probability close to 1 with $\mathcal{O}(\sqrt{N})$ oracle calls, where N is the size of the database. Thus, Grover's routine provides a quadratic speedup over classical methods [2], [3].

The contribution of [4] extended the scope of Grover's algorithm to the search for a minimum, the so-called Grover adaptive search. Consequently, the algorithm provides an alternative quantum approach to existing algorithms, such as the Quantum Approximate Optimization Algorithm and the variational quantum eigensolver [5], [6], to address to problems like quadratic unconstrained binary optimization and constrained polynomial binary optimization [7]. More generally, this opens up a wide range of possibilities for using Grover's algorithm to solve combinatorial optimization problems in various technological and industrial fields.

For instance, Grover's algorithm has already been applied to the graph coloring problem [8], [9] and AES key search [10]. Several studies have also adapted the algorithm to various signal processing tasks. Among the existing works on this topic, one can mention [11], which adapts Grover's search to image pattern matching, and [12], which uses the algorithm in the context of wireless communications to compute a maximum-likelihood detector.

Even though Grover's algorithm provides a quadratic speedup, it remains a relevant issue to optimize its computational cost. Indeed, the claimed quadratic speedup does not account for the cost of an oracle call during inside a single iteration [13]. However, the resources required for each iteration, such as the quantum gates that implement the so-called Grover's oracle, can rapidly increase depending on the criterion of the search problem [14]. Existing works [15], [16] propose directions for optimizing the circuit implementation of the algorithm, but it may also be promising to optimize the number of oracle calls of Grover's algorithm. For example, Cherckesova et al. [17] proposed modifying the initialization of Grover's procedure to save several iterations. Nevertheless, there are still various strategies to explore in order to further reduce the number of Grover's iterations without compromising its reliability.

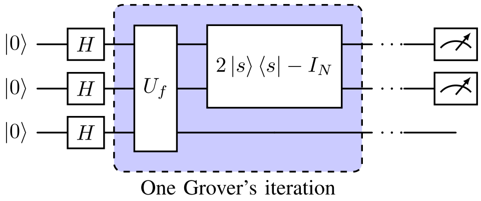


FIGURE 1. Grover scheme with for 2 qubits. The last qubit is an ancilla qubit needed for marking the solutions.

This work aims to optimize the number of iterations of Grover's algorithm by using a hybrid approach. Inspired by the BBHT approach [2] that launches Grover's algorithm several times, we propose to introduce one additional degree of freedom to the usual number of iterations of the algorithm, called the number of trials. This parameter permits to run Grover several times if needed, allowing to reduce the number of iterations within one Grover's trial. In this work, we analyze the performances of such algorithm. The rest of this article is organized as follows. First, we review the description of Grover's algorithm in Section II before describing our proposal in Section III. We discuss how to properly parameterize our hybrid algorithm in Section V and we analyze its performance in Section VI. Finally, Section VII concludes this article.

II. GROVER'S ALGORITHM

A. DESCRIPTION OF GROVER'S ROUTINE

Assume that we have a database with $N = 2^n$ unsorted elements labeled by a bit-string $x \in \{0,1\}^n$, and a Boolean function $f : \{0,1\}^n \to \{0,1\}$ that marks the elements x_0 contained in a subset $\mathcal{S}$ of $\{0,1\}^n$, such that

$$f(x) = \begin{cases} 1, & \text{if } x = x_0 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Grover's algorithm [1], [2] aims to find one of these marked elements. It consists in a series of gates applied on a quantum register composed of n qubits, in order to amplify the amplitude of the targeted elements. This way, the probability to measure one of the marked states is close to 1 at the end of the computation. The algorithm relies on the fact that one is able build a unitary operator U_f, often called oracle, such that

$$U_f x = \begin{cases} -|x\rangle, & \text{if } f(x) = 1 \\ |x\rangle, & \text{if } f(x) = 0. \end{cases} \tag{2}$$

Of course, U_f depends on the criteria used to define the subset $\mathcal{S}$. In this work, we assume that the oracle is well defined and we consider the behavior of the algorithm independently of U_f.

Let us denote $|\Psi\rangle \in \mathcal{H}^{\otimes n}$ the state of the register. Grover's routine consists in the procedure described in Algorithm 1. The instructions within the while block constitute one Grover's iteration, depicted on the quantum circuit of Fig. 1.

```
Algorithm 1: Grover's Algorithm.
Input: $k \in \mathbb{N}^*$
Initialize the register $|\Psi\rangle$ to the uniform
superposition: $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle$
for $i = 1$ to $k$ do
    Apply the oracle $U_f$
    Apply the diffusion operator $D = 2|s\rangle\langle s| - I_N$
end for
Perform a measure of the register
Output: $|x\rangle$
```

At the beginning, all the states of the computational basis are equiprobable. Applying the oracle U_f marks the elements of $\mathcal{S}$. Then, the diffusion operator amplifies the marked elements, while attenuating the amplitude of the other ones. Assume that t elements are solutions, i.e., $|\mathcal{S}| = t$. After k iterations, it is shown in [2] that the state of the register reads

$$|\Psi\rangle = \left(DU_f\right)^k |s\rangle = \frac{1}{\sqrt{t}} \sin\left((2k+1)\theta_G\right) \sum_{x_0 \in S} |x_0\rangle + \frac{1}{\sqrt{N-t}} \cos\left((2k+1)\theta_G\right) \sum_{x \notin S} |x\rangle \tag{3}$$

where the angle θ_G is defined by

$$\sin^2(\theta_G) = \frac{t}{N}. \tag{4}$$

B. SUCCESS PROBABILITY WITHIN ONE SHOT

A measure of $|\Psi\rangle$ after k iterations makes the state collapse to one of the marked elements with the probability

$$P_{\mathrm{BG}}(k) = \sin^2((2k+1)\theta_G) \tag{5}$$

which corresponds to the success probability of Grover's algorithm after k iterations. The optimal number of iterations one should perform to reach the highest success probability with one shot of Grover's algorithm is chosen so that (5) is maximized. It is usually denoted L_{opt} and given by [2]

$$L_{\mathrm{opt}} = \left\lfloor \frac{\pi}{4} \sqrt{\frac{N}{t}} \right\rfloor. \tag{6}$$

The algorithm costs $O(\sqrt{N})$ iterations, hence one often mentions a quadratic speedup. The case $t > 1$ where multiple elements are marked raises the issue of the choice of L_{opt} when $\mathcal{S}$ is not well known as discussed in [2] and [18]. In this work, we choose to focus on the case $t = 1$. Indeed, notice that all the observable quantities depend on the ratio N/t. Hence one can recover the multisolution case by rescaling $N \to N/t$ when t is known.

C. INDEPENDENT REPETITIONS

One might consider another strategy to parameterize Grover's algorithm in order to ensure a success rather than

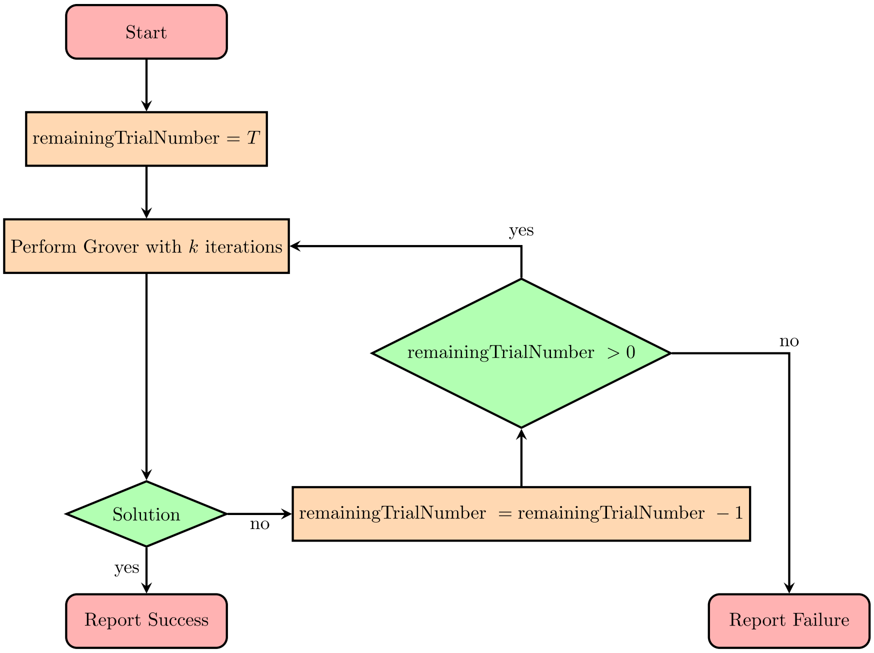


FIGURE 2. **Proposed searching scheme for reducing the number of iterations.**

settling for a finite success probability. This can be achieved by allowing multiple executions of the previous procedure, but the parameterization of the algorithm must be slightly adjusted. Indeed, once the number of iterations of Grover's algorithm is fixed to a certain k, each repetition of the routine corresponds to a Bernoulli trial with a success probability $P_{\mathrm{BG}}(k)$.

Assuming unbounded computational resources, the number of shots required to achieve one success follows a geometric distribution with parameter $p = P_{\mathrm{BG}}(k)$. The expected number of shots to achieve a success is given by

$$\langle n_{\mathrm{shot}} \rangle = \sum_{j=1}^{\infty} j\,(1 - P_{\mathrm{BG}}(k))^{j-1}\, P_{\mathrm{BG}}(k) = \frac{1}{P_{\mathrm{BG}}(k)} \tag{7}$$

which yields a mean running time, expressed in terms of oracle calls, as $k/P_{\mathrm{BG}}(k)$. Then, if one denotes $\mathsf{k_0}$ the minimizer of this running time, i.e.,

$$\mathsf{k_0} = \underset{k\in\mathbb{R}^+}{\arg\min} \left\{ \frac{k}{P_{\mathrm{BG}}(k)} \right\} \tag{8}$$

the optimal number of iteration within one Grover's routine appears to be instead

$$L_{\mathrm{opt}}^{(\infty)} = \lfloor \mathsf{k_0} \rfloor. \tag{9}$$

However, this analysis assumes that Grover's algorithm can be repeated indefinitely, which is not the case in practice due to limited computational time on quantum processors [19]. It is why one usually accommodates with success probabilities of quantum algorithms strictly under 100% [20] even with several repetitions.

Nevertheless, running Grover's routine with a maximal number of shots may be useful, as its success probability can be tuned by adjusting the number of iterations within each trial. This work aims to explore the potential benefits of such parameterization.

III. MIXED GROVER (MG) PROPOSAL

In this section, we propose investigating a novel strategy to run Grover's algorithm. As mentioned previously, a perfect success probability of 100% is unattainable in practice. Instead, we aim to find a process that consumes as few iterations as possible while achieving a targeted success.

A. PROPOSED ALGORITHM

Let us denote BG the basic Grover's algorithm for further convenience. The latter corresponds to the natural approach where the algorithm is run only once with the optimum number of iterations $k = L_{\mathrm{opt}}$ given by (6) [1]. Reducing the parameter k speeds up the computation but at the cost of performance degradation.

Nonetheless, we propose a new strategy illustrated on Fig. 2. It consists in running Grover's algorithm with k iterations where $k < L_{\mathrm{opt}}$ deliberately. The output $|x\rangle$ provided by Grover is tested with a classical algorithm, which evaluates if the output effectively verifies $f(x) = 1$. If it is validated, the computation stops and thus saves unnecessary iterations. Otherwise, a new trial is launched. However, contrarily to Section II-C, we limit the maximum number of shots to a certain number T to bound the algorithm resource consumption. For further concision, we simply call T the number of trials but it corresponds in fact to an upper bound on the number of shots of Grover's routine. The algorithm is then parameterized with a certain pair of integers (k, T). We present the procedure in Algorithm 2.

Algorithm 2: MG Algorithm.

```
Input: $T \in \mathbb{N}^*, k \in \mathbb{N}^*$
trialNumber $\Leftarrow T$
while trialNumber $> 0$ do
    Perform Grover's algorithm (Algorithm 1) with $k$ iterations
    Test classically if the output $|x\rangle$ verifies $f(x) = 1$
    if Success then
        trialNumber $\Leftarrow 0$
    else
        trialNumber $\Leftarrow$ trialNumber - 1
    end if
end while
Output: $|x\rangle$
```

This algorithm is denoted by MG as both classical and quantum computations are used. It must be noted that, to provide an output for the classical algorithm, the quantum part makes a measurement that suppresses the states superposition. Hence, at each new trial, one cannot exploit the previous trials, and the Grover algorithm has to start from scratch.

With this hybrid stratagem, we are looking for configurations that would cost less computational time to reach a given success probability.

B. DEFINITION OF THE METRICS

In order to determine whether this mixed approach presents an advantage compared to BG, one needs to define appropriate metrics to reflect the obtained performance. As we would like to quantify the reliability and the computational cost of our approach, we have to consider two associated criteria in this work. The first one is the success probability, and the second one is the mean number of iterations.

Let us begin with the success probability. It is obtained by considering the success rates of all Grover's shots up to Tth one.

Proposition 1 (Success Probability of MG): The success probability P_{MG} of the MG algorithm is given by

$$P_{\mathrm{MG}}(k, T) = 1 - (1 - P_{\mathrm{BG}}(k))^T. \tag{10}$$

Proof: At each shot of k iterations, Grover's routine returns a solution with the probability $P_{\mathrm{BG}}(k)$ as defined in (5). Hence the probability to retrieve the solution after exactly j BG shots is $(P_{\mathrm{BG}}(k)) \times (1 - P_{\mathrm{BG}}(k))^{j-1}$. The formula of (10) is obtained by summing these probabilities up to the number of trials T.

The complexity of MG's algorithm is evaluated with the expected number of shots multiplied by the number of iterations within one Grover's routine. Thus, it corresponds to the mean number of oracle calls used by MG. We define the metric E_i as the expected number of iterations in order to evaluate the complexity of MG.

Proposition 2 (Expected Number of Iterations): The expected number of iterations performed by MG is given by

$$E_i(k, T) = \frac{k}{P_{\mathrm{BG}}(k)} P_{\mathrm{MG}}(k, T). \tag{11}$$

Proof: The expected number of Grover shots within MG have the two following contributions:

1) expected number of shots to get a success with at most $T - 1$ trials, which corresponds to the sum of (7) truncated to the first $T - 1$ terms;
2) worst case where the T trials are consumed, which occurs with the probability $(1 - P_{\mathrm{BG}}(k))^{T-1}$.

Thus, the expected number of shots effectively used within MG is given by the sum

$$\sum_{j=1}^{T-1} j\,(1 - P_{\mathrm{BG}}(k))^{j-1}\,P_{\mathrm{BG}}(k) + T(1 - P_{\mathrm{BG}}(k))^{T-1}$$
$$= \frac{1 - (1 - P_{\mathrm{BG}}(k))^T}{P_{\mathrm{BG}}(k)}. \tag{12}$$

Using the expression of P_{MG} and multiplying by k (the number of iterations within one Grover shot) gives the expression of (11).

As a sanity check, we can inspect the behavior of our metrics in the $T \to \infty$ limit. Given a certain number of Grover's iterations k, one has $0 < P_{\mathrm{BG}}(k) < 1$. Thus, it straightforward to see that

$$\begin{cases} P_{\mathrm{MG}}(k, T) & \xrightarrow[T\to\infty]{} 1 \\ E_i(k, T) & \xrightarrow[T\to\infty]{} \frac{k}{P_{\mathrm{BG}}(k)}. \end{cases} \tag{13}$$

One recovers the mean number of repetitions of BG with k iterations to get a success with certainty, which is consistent. We can consider that the MG proposal includes in fact this asymptotic case, as it is approached with large values of T.

Let us now return to the case of interest in this work, namely, a finite T, to account for bounded computational resources.

C. PARAMETERIZATION OF THE ALGORITHM

The initialization of the MG algorithm requires a wise choice of the parameters k and T. Indeed, one has to find a compromise between the reliability P and the complexity E. This problem can be seen from two different angles as follows.

1) One can maximize the success rate of the algorithm given a complexity constraint, which corresponds to the problem P1

$$\text{Solve} \quad \underset{(k,T)\in(\mathbb{N}^*)^2}{\arg\max} \; \{P_{\mathrm{MG}}(k, T) \text{ s.t } E_i(k, T) \le E\}. \tag{P1}$$

2) Otherwise, one might minimize the complexity under a reliability constraint. It amounts in solving the problem

P2

$$\text{Solve} \quad \underset{(k,T)\in(\mathbb{N}^*)^2}{\arg\min} \quad \{E_i(k,T) \ \text{s.t} \ P_{\mathrm{MG}}(k,T) \geq p\}. \qquad \text{(P2)}$$

Depending on the user's requirements, one might consider P1 or P2. The first one permits to obtain the best performances with a constrained allowed conplexity, while the second one tends to reduced the complexity for a targeted reliability.

IV. FIRST EVIDENCE OF MG'S ADVANTAGE

In this section, we show that MG presents an interest over current implementations of Grover's algorithm. We first discuss it in the light of the P1 formulation, and we mention the P2 parameterization as well.

A. CURRENT METHODS FOR DEALING WITH COMPLEXITY CONSTRAINTS

As discussed previously, the Grover's algorithm could be used according to one of those two current methods as follows.

1) One can only use one shot of Grover's algorithm (BG). In this case, we can set the number of iterations to $k = L_{\text{opt}}$ to maximize the success rate.
2) One can use unlimited number of repetitions of Grover's algorithm (Infinite Grover: IG). In this case, we can set the number of iterations within one shot to $k = L_{\text{opt}}^{(\infty)}$ to minimize the mean running time and repeat the algorithm until success.

However, these methods must be adapted if one wants to reduce the total number of oracle calls to a certain value E. In the one shot case (i.e., the traditional use of Grover's algorithm), the number of iterations k must be lowered until $k \leq E$ which will degrade the success probability $P_{\mathrm{BG}}(k)$. On the other hand in the second scenario, one can still run Grover's routine several times with $L_{\text{opt}}^{(\infty)}$ iterations within each trial, but then limit the number of shots to not exceed E oracle calls. In both cases, the success probability of the searching procedure falls strictly under 100%. We can define the success rate of the current methods limited to E oracle calls as the highest value between the two following ones.

1) The success probability of a single Grover's routine using E oracle calls.
2) The success probability after repeating Grover's routine parameterized with $L_{\text{opt}}^{(\infty)}$ iterations using up to E oracle calls on average.

Alternatively, the parameterization of MG will provide new configurations that are not considered by current methods. This could lead to different suggestions for the number of iterations k within a single Grover's routine and the maximal number of repetitions T of BG, potentially achieving better compromises.

B. INTEREST OF MG

We have seen previously that the running time of Grover's algorithm to find a marked element with certainty is given by $L_{\text{opt}}^{(\infty)}/P_{\mathrm{BG}}\left(L_{\text{opt}}^{(\infty)}\right)$. If the maximum allowed oracle calls E exceeds this value, one can simply repeat Grover's algorithm with $L_{\text{opt}}^{(\infty)}$ (IG) without seeking alternative strategies. Then, let us assume that one has a limited number of oracle calls $E \leq L_{\text{opt}}^{(\infty)}/P_{\mathrm{BG}}\left(L_{\text{opt}}^{(\infty)}\right)$.

For concreteness, consider $n = 10$ (giving a database size of $N = 2^n = 1024$) where

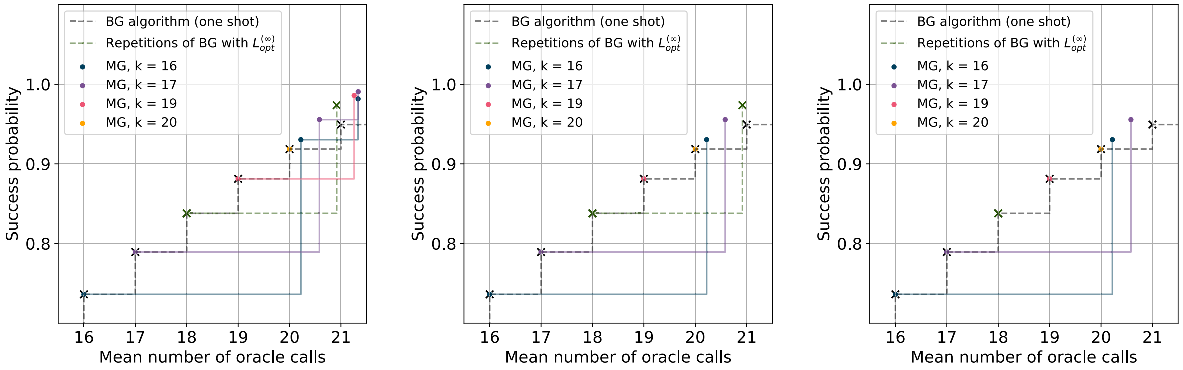
$$\begin{cases} L_{\text{opt}}^{(\infty)} & = 18 \\ L_{\text{opt}}^{(\infty)}/P_{\mathrm{BG}}\left(L_{\text{opt}}^{(\infty)}\right) & \approx 21.48. \end{cases}$$

We first focus on MG's advantage for maximizing the success rate under a complexity constraint, as adressed by the P1 problem. We depicted on Fig. 3 the success rate against the complexity cost of the MG algorithm for several parameterizations (k, T). A given color corresponds to a fixed number of iterations k within one Grover's shot, and each additional point represents the obtained performances by allowing one additional trial. On the other hand, the current methods corresponds to either running BG with E iterations once; or repeating BG with $L_{\text{opt}}^{(\infty)}$ iterations within each shot. The configurations corresponding to current methods are represented with crosses on the plot, while the MG configurations correspond to dots.

Each plot depicted on Fig. 3 corresponds to a specific value of the maximal number of oracle calls E. Configurations that exceed this limit are not shown. Consequently, the best parameterization in each plot is the one that maximizes the success probability. The results presented in Fig. 3(b) suggest that, for the corresponding values of E, the optimal configuration is actually a current method of implementing Grover's algorithm. However, the conclusion appears to be different in Fig. 3(a) and (c). Indeed, these results show, on the contrary, that certain MG configurations outperform the current methods.

We can then claim that MG presents an advantage over standard implementations of Grover's algorithm for maximizing the success rate given a maximal number of oracle calls, for certain values of the upper bound E. In the example shown above, the region of interest for MG is relatively small, but is expected to expand as the database size increases. This suggests that MG could become increasingly beneficial in practical scenarios where larger search spaces are involved.

In addition, one can also assess the MG's advantage of reducing the complexity under a success rate constraint, as adressed by the P2 problem. One can observe on Fig. 3(b) that, for a similar success probability, MG needs less oracle calls than the current methods. Indeed, MG reaches a success probability of 85% with 20.55 mean number of oracle calls, while BG achieve approximately the same success rate (even slightly smaller) but with 21 oracle calls.



(a)

(b)

(c)

FIGURE 3. Parametric plot of $(E_i(k, T), P_{\mathrm{MG}}(k, T))$ for several pairs (k, T) ($n = 10$, $N = 2^{10}$). For multiple values of E, MG configurations provide better success rates than current methods (a), (c). There are also some values of E where the best configuration aligns with a current method (b). (a) $E = 21.35$. (b) $E = 21.25$. (c) $E = 20.6$.

Thus, MG also presents an advantage over standard implementations of Grover's algorithm for minimizing the number of oracle calls given a targeted success rate. Actually, one may note that P1 and P2 are dual. Indeed, a compromise has to be made between complexity and reliability, so reducing the number of oracle calls for a targeted success rate as in P2 will permit to increase the success rate for a constrained complexity as in P1.

In this article, we chose to focus on the potential benefits of the alternative parameterization approach P2 for MG.

V. THEORETICAL ANALYSIS

In this section, we would like to go further in the reduction of oracle calls for Grover's searching problems. To do so, we propose to focus on the P2 formulation of the parameterization of MG. Given a reliability constraint, we aim to minimize the number of oracle calls.

We define the optimal pair (k^*, T^*) for MG as the one that permits to minimize the complexity while satisfying such constraint

$$(k^*, T^*) = \underset{(k,T)\in(\mathbb{N}^*)^2}{\arg\min} \; \{E_i(k, T) \text{ s.t. } P_{\mathrm{MG}}(k, T) \geq p\}. \tag{14}$$

In order to find the exact solution with perfect accuracy, one could perform an exhaustive search on k and T, which would cost a lot of time. In this section, we elaborate a generic procedure to get a pair of integers $(k_{\mathrm{MG}}, T_{\mathrm{MG}})$ which approximates the solution. To do so, we will study first the continuous version of this problem and construct our formula from its solution.

A. DEFINITION OF THE SET OF PARAMETERS

According to (14), the optimal pair is in the set

$$I_p = \left\{(k, T) \in \left(\mathbb{N}^*\right)^2 \mid P_{\mathrm{MG}}(k, T) \geq p\right\}. \tag{15}$$

One can be more precise by giving additional restrictions. Indeed, it can be shown that for $T \geq 1$

$$E_i(k, T) \geq k. \tag{16}$$

Thus, MG cannot provide any complexity advantage over BG if it is configured with a number of iterations greater than L_{opt}. Hence, we bound the parameter k with L_{opt}, so that the set of parameters is reduced to $I_p \bigcap([1, L_{\mathrm{opt}}] \times \mathbb{N}^*)$

$$I_p^{(\mathbb{N}\times\mathbb{N})} = \left\{(k, T) \in \{1, \ldots, L_{\mathrm{opt}}\} \times \mathbb{N}^* \mid P_{\mathrm{MG}}(k, T) \geq p\right\}. \tag{17}$$

We provide a representation of this set in the plane (k, T) on Fig. 4(a). Solving this optimization problem in a discrete space is challenging. We first propose addressing a relaxed version of the problem defined over a continuous space, and then identifying the discrete solution.

B. CONTINUOUS CASE AND CRITICAL PROBABILITY

We first deal with the problem of (14) by considering the continuous case, i.e., by relaxing the constraint on the parameters to $(k, T) \in [1, L_{\mathrm{opt}}] \times [1, +\infty)$. Let us denote $I_p^{(\mathbb{R}\times\mathbb{R})}$ the natural extension of the set defined in (17)

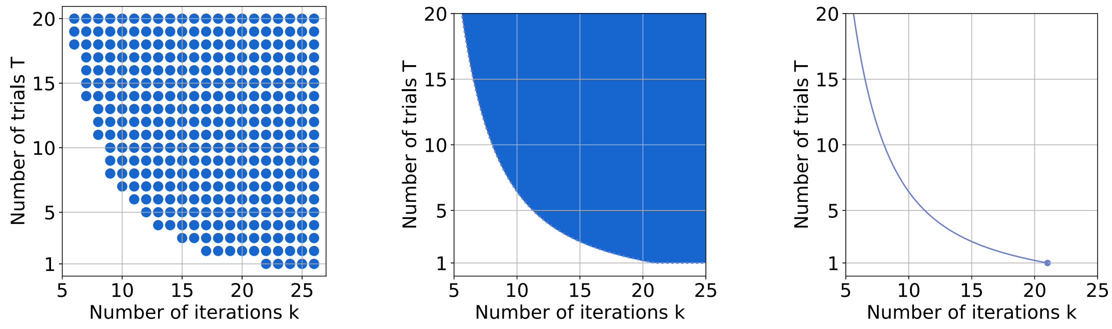
$$I_p^{(\mathbb{R}\times\mathbb{R})} = \{(k, T) \in [1, L_{\mathrm{opt}}] \times [1, +\infty) \mid P_{\mathrm{MG}}(k, T) \geq p\}. \tag{18}$$

This continuous extension is represented on Fig. 4(b). Our initial problem now reads

$$\min_{(k,T)\in I_p^{(\mathbb{R}\times\mathbb{R})}} E_i(k, T) \tag{19}$$

and it can be solved using the following theorem.

Theorem 1: The minimization problem (19) admits a unique solution $(\mathrm{k}^{(1)}, \mathrm{T}^{(1)})$ for any $p \in (0, 1)$ located on the boundary $I_p^{(1)} = I_p^{(\mathbb{R}\times\mathbb{R})} \bigcap\{(k, T), P_{\mathrm{MG}}(k, T) = p\}$ shown



(a)

(b)

(c)

FIGURE 4. **Visualization of the different sets $I_p^{(\mathbb{N}\times\mathbb{N})}$, $I_p^{(\mathbb{R}\times\mathbb{R})}$ and $I_p^{(1)}$ for $n = 10$ (where $L_{\text{opt}} = 25$) and $p = 0.95$. (a) Set $I_p^{(\mathbb{N}\times\mathbb{N})}$. (b) Set $I_p^{(\mathbb{R}\times\mathbb{R})}$. (c) Set $I_p^{(1)}$.**

on Fig. 4(c). The optimal pair is given by

$$(\mathsf{k}^{(1)}, \mathsf{T}^{(1)}) = \begin{cases} \left(\mathsf{k}_0, \frac{\log(1-p)}{\log(1-p_c)}\right), & \text{if } p > p_c \\ \left(\frac{1}{2}\left(\frac{1}{\theta_G}\arcsin(\sqrt{p}) - 1\right), 1\right), & \text{if } p \leq p_c \end{cases} \tag{20}$$

where $\mathsf{k}_0 \in [1, L_{\text{opt}}]$ minimizes the ratio $k/P_{BG}(k)$. In particular, one has

$$\tan((2\mathsf{k}_0 + 1)\theta_G) = 4\mathsf{k}_0\theta_G \tag{21}$$

and the critical probability p_c is defined by

$$p_c = \frac{16\mathsf{k}_0^2\theta_G^2}{1 + 16\mathsf{k}_0^2\theta_G^2}. \tag{22}$$

Proof: See Appendix A

For the sake of clarity, the solution is denoted with the mathematical font $(\mathsf{k}^{(1)}, \mathsf{T}^{(1)})$ to remind the reader that it might be a nonphysical configuration since MG cannot be parameterized with noninteger inputs. We keep this convention for the rest of this article. Besides, one may note that $\left(\mathsf{k}^{(1)}, \mathsf{T}^{(1)}\right)$ depends on the pair (p, n). However, this dependency will be omitted to simplify the notation.

Let us first comment the obtained solution in the case $p > p_c$. The parameter $\mathsf{k}^{(1)}$ does not actually depend on the target success probability, which might seem surprising. It is given by k_0, that is the number of oracle calls that minimizes the ratio $k/P_{\text{BG}}(k)$. One recognizes from (7) that this ratio corresponds to the mean running time of Grover's algorithm to find a solution, with k iterations and an unlimited number of shots. Thus, the construction of the solution of the continuous problem for $p > p_c$ can be seen as a two-step process as follows.

1) Set the number of oracle calls such that the mean running time of Grover's algorithm to obtain a success is optimal.
2) Adapt the number of shots to the desired reliability $p < 100\%$.

This first approach also underlines that for p below the critical value p_c, MG has no chance to provide any advantage compared to BG since $\mathsf{T}^{(1)} = 1$. So from now on, we focus on the region $p > p_c$.

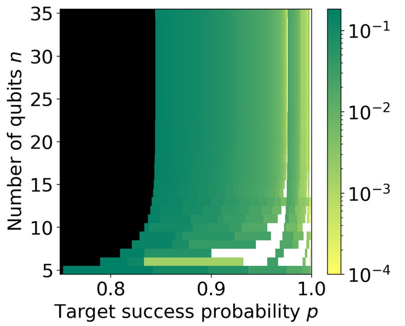


FIGURE 5. **Relative error $(E_i^{\text{MG}} - E_i^*)/E_i^*$ between the configuration $(k_{\text{MG}}, T_{\text{MG}}) = (\lceil\mathsf{k}^{(1)}\rceil, \lceil\mathsf{T}^{(1)}\rceil)$ and the true optimal configuration. The black region corresponds to $p < p_c$.**

Starting from this result, one could be tempted to convert the real values into integer ones to configure the algorithm. It yields the following proposal:

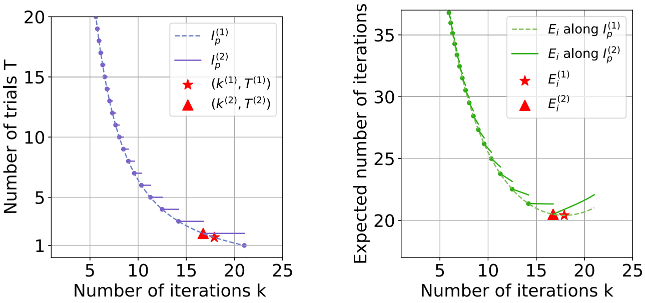
$$(k_{\text{MG}}, T_{\text{MG}}) \overset{?}{=} (\lceil\mathsf{k}^{(1)}\rceil, \lceil\mathsf{T}^{(1)}\rceil) \tag{23}$$

where the upper integer part $\lceil.\rceil$ is the most suitable because P_{MG} is monotonically increasing on $[1, L_{\text{opt}}]$ (respectively, on $[1, +\infty)$) at fixed T (respectively, fixed k). However, Fig. 5 shows the relative error between the expected number of iterations one obtains with this configuration and what one gets with the optimal configuration (k^*, T^*) found by an exhaustive search. The black part corresponds to the case where $p < p_c$, ie. where BG is already the best strategy. We can observe that there is only a small part of the cases where (23) leads to the actual optimum (white area). Otherwise, the relative error can exceed 10%. Thus, we derive in the next parts a new strategy.

C. PARAMETERS DISCRETIZATION

1) NUMBER OF TRIALS

Starting from the solution for continuous case, we treat an intermediate problem where we discretize the parameter T and keep the variable k continuous. For convenience, we



(a)

(b)

FIGURE 6. Visualization of $I_p^{(1)}$ and $I_p^{(2)}$ (left) and the objective $E_i(k, T)$ along them (right) for $n = 10$ qubits (where $L_{\text{opt}} = 25$) and $p = 0.95$. $E_i^{(1)}$ (resp. $E_i^{(2)}$) is the minimum of E_i along $I_p^{(1)}$ (resp. $I_p^{(2)}$). In this case, the minimum $E_i^{(2)}$ is located precisely at $k = u[m]$. (a) Parametric curves $I_p^{(1)}$ and $I_p^{(2)}$. (b) Different shapes of E_i.

introduce the following parametric representation:

$$\{(k, T) \text{ s.t } P_{\text{MG}}(k, T) \geq p\} = \{(k, T(k, p))\} \tag{24}$$

where we define $T(k, p)$ as

$$T(k, p) = \frac{\log(1 - p))}{\log(1 - P_{\text{BG}}(k))}. \tag{25}$$

Consequently, the set of parameters $I_p^{(\mathbb{R}\times\mathbb{N})}$ is given by

$$\begin{aligned} I_p^{(\mathbb{R}\times\mathbb{N})} &= \{(k, T) | 1 \leq k \leq L_{\text{opt}} \\ T &= \lceil T(k, p)\rceil, \lceil T(k, p)\rceil + 1, \ldots\}. \end{aligned} \tag{26}$$

The upper integer part $\lceil\cdot\rceil$ comes from the monotony of P_{MG} at fixed k that we already mentioned.

Inspired by the continuous case, we choose to optimize the expected number of iterations along the boundary of $I_p^{(\mathbb{R}\times\mathbb{N})}$ that we denote $I_p^{(2)}$

$$I_p^{(2)} = \left\{(k, \lceil T(k, p)\rceil) \mid 1 \leq k \leq L_{\text{opt}}\right\}. \tag{27}$$

This set is a discretization of $I_p^{(1)}$ along T. Thus, as can be seen on Fig. 6(a), it corresponds to a new parametric curve with a piecewise structure where each branch is associated to an integer value of $T = \lceil T(k, p)\rceil$.

The objective is now to find the pair $(\mathsf{k}^{(2)}, T^{(2)})$ along $I_p^{(2)}$ that minimizes E_i. This optimization procedure is more tricky since E_i has also now a piecewise structure. Indeed, the representative curve of E_i along $I_p^{(2)}$ denoted $E_i|_{I_p^{(2)}}$ has several branches that deviate from $E_i|_{I_p^{(1)}}$, as shown on Fig. 6(b). Nevertheless, we can determine a region where the minimum should be located. To do so, one defines the sequence $(u[j])_{j\geq 1}$ by

$$u[j] = \frac{1}{2}\left(\frac{1}{\theta_G}\arccos\left((1 - p)^{1/2j}\right) - 1\right) \tag{28}$$

whose successive terms delimit the regions where $\lceil T(k, p)\rceil$ is constant, corresponding to the branches of $E_i|_{I_p^{(2)}}$. We defer

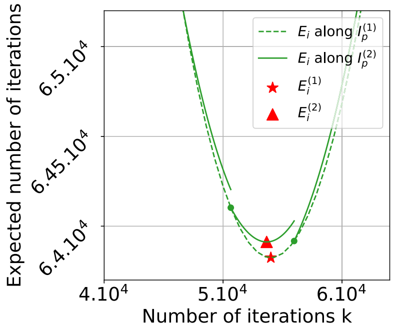


FIGURE 7. Example where the minimum $E_i^{(2)}$ is located inside the branch of interest, i.e., $u[m] < \mathsf{k}^{(2)} < u[m - 1]$ (obtained with $n = 33$ qubits, $p = 0.998$).

the formal justification to Appendix B. Let us denote

$$m = \lceil \mathsf{T}^{(1)}\rceil. \tag{29}$$

The terms $u[m]$ and $u[m - 1]$ allow to define a region corresponding to the branch of $E_i|_{I_p^{(2)}}$ we should focus on, as underlines the following theorem.

Theorem 2: For $p > p_c$, the expected number of iteration E_i admits a unique minimizer $(\mathsf{k}^{(2)}, T^{(2)})$ on $I_p^{(2)}$. It is located on the mth branch or eventually on the boundary of the $m - 1$th branch

$$\begin{cases} u[m] \leq \mathsf{k}^{(2)} < u[m - 1] & T^{(2)} = m \\ \text{or} & \\ \mathsf{k}^{(2)} = u[m - 1] & T^{(2)} = m - 1. \end{cases} \tag{30}$$

In the case $u[m] < \mathsf{k}^{(2)} < u[m - 1]$, the latter verifies

$$\begin{aligned} &\tan((2\mathsf{k}^{(2)} + 1)\theta_G) = 4\mathsf{k}^{(2)}\theta_G \\ &\quad - 4\mathsf{k}^{(2)}\theta_G \frac{mP_{\text{BG}}(\mathsf{k}^{(2)})(1 - P_{\text{BG}}(\mathsf{k}^{(2)}))^{m-1}}{P_{\text{MG}}(\mathsf{k}^{(2)}, m)}. \end{aligned} \tag{31}$$

Proof: See Appendix C.

The pair $(\mathsf{k}^{(2)}, T^{(2)})$ provides an optimal MG configuration in $[1, L_{\text{opt}}] \times \mathbb{N}^*$. In the case depicted on Fig. 6, the pair is located exactly on the boundary of the mth branch, ie. $\mathsf{k}^{(2)} = u[m - 1]$. However, the example depicted in Fig. 7 shows that there also exist cases where $u[m] < \mathsf{k}^{(2)} < u[m - 1]$.

The next and last step is to discretize the number of Grover's iterations k within one trial.

2) NUMBER OF ITERATIONS WITHIN ONE GROVER SHOT

The final operation to perform is to take the upper integer part of $\mathsf{k}^{(2)}$. Hence, we propose the configuration

$$(k_{\text{MG}}, T_{\text{MG}}) = (\lceil \mathsf{k}^{(2)}\rceil, T^{(2)}). \tag{32}$$

However, there is a mathematical subtlety. The discretization $\mathsf{k}^{(2)} \longrightarrow \lceil \mathsf{k}^{(2)}\rceil$ increases the expected number of iterations. Because of the piecewise structure of E_i along $I_p^{(2)}$, we cannot be sure that an integer part obtained from a local minimum on another branch would not be better. We expect that to happen

Algorithm 3: MG Configuration.

```
Input: $n \in \mathbb{N}^*$, $p > 0$
Find the solution $\mathsf{k}_0$ in $[1, L_{\text{opt}}]$ of the equation:
    $\tan((2k+1)\theta_G) = 4k\theta_G$
Compute the critical probability:
    $p_c = \frac{16\mathsf{k}_0^2\theta_G^2}{1+16\mathsf{k}_0^2\theta_G^2}$
if $p \leq p_c$ then use BG
    $k_{MG} = \left\lceil \frac{1}{2}(\frac{1}{\theta_G}\arcsin(\sqrt{p}) - 1) \right\rceil$
    $T_{MG} = 1$
else
    $m \Leftarrow \left\lceil \frac{\log(1-p)}{\log(1-p_c)} \right\rceil$
    $k_{inf} \Leftarrow u[m]$
    $k_{mid} \Leftarrow +\infty$
    $k_{sup} \Leftarrow u[m-1]$
    Check the eventual solutions in $[k_{inf}, k_{sup})$ of:
        $\tan((2k+1)\theta_G) = 4k\theta_G - 4k\theta_G \frac{mP_{\text{BG}}(k)(1-P_{\text{BG}}(k))^{m-1}}{P_{\text{MG}}(k,m)}$
    if a solution exists then
        $k_{mid} \Leftarrow$ found solution
    end if
    $(\mathsf{k}^{(2)}, T^{(2)}) \Leftarrow \text{argmin}\{E_i(k_{inf}, m), E_i(k_{mid}, m), E_i(k_{sup}, m-1)\}$
    $k_{MG} = \lceil \mathsf{k}^{(2)} \rceil$
    $T_{MG} = T^{(2)}$
end if
Output: $(k_{MG}, T_{MG})$
```

only at low number of qubits where the width of the branches of $E_i|_{I_p^{(2)}}$ remains small.

D. PROPOSAL: SUMMARY AND RELIABILITY

To sum up, we give in Algorithm 3 the steps to build the final MG configuration as defined in (32).

Let us now analyze the deviation between the real integer optimum for MG and our proposal. We compare again the performance of the true integer optimum (k^*, T^*) and this proposal ones. Fig. 8 shows the relative error between the expected number of iterations in the configuration of (32) and the integer optimum E_i^* for a number of qubits n comprised in $\{5, \ldots, 35\}$. The white zone corresponds to perfect 0's. Except for a small difference at low n (but quantum algorithms are more intended for higher database size), (32) permits to find the true integer optimum, leading to a much better reliability than the one obtained on Fig. 5. Thus, our approach to parameterize MG is appropriate, and exact for $n \geq 7$.

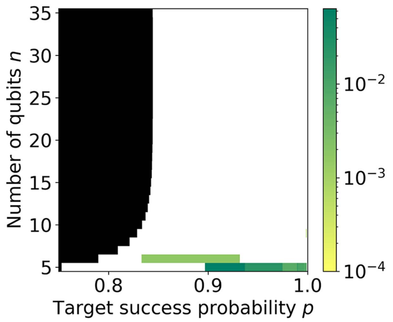


FIGURE 8. Relative error $(E_i^{\text{MG}} - E_i^*)/E_i^*$ between the configuration $(k_{\text{MG}}, T_{\text{MG}}) = (\lceil \mathsf{k}^{(2)} \rceil, T^{(2)})$ and the optimal configuration. The black region corresponds to $p < p_c$.

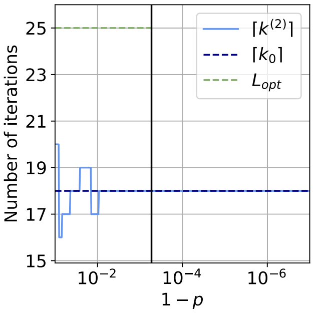


FIGURE 9. Convergence of $\lceil \mathsf{k}^{(2)} \rceil$ toward $\lceil \mathsf{k}_0 \rceil$ as the targeted success p increases with $n = 10$ qubits. We also indicate the number of iterations L_{opt} of BG. There is no data for $p < p_c$.

E. BEHAVIOR FOR HIGH TARGET SUCCESS PROBABILITY

A higher resolution of Fig. 5 would show that there exists a tiny white region for a targeted success p close to 1 where the configuration $(\lceil \mathsf{k}^{(1)} \rceil, \lceil \mathsf{T}^{(1)} \rceil)$ corresponds in fact to the optimal pair for MG. It comes from the following theorem.

Theorem 3: As $p \to 1$, one has the following convergence property:

$$\mathsf{k}^{(2)} \underset{p \to 1}{\longrightarrow} \mathsf{k}_0. \tag{33}$$

Proof: See Appendix D.

Consequently, one could expect that $\lceil \mathsf{k}^{(2)} \rceil$ might be replaced by $\lceil \mathsf{k}_0 \rceil$ for p very close to 1. Again some subtleties need to be mentioned. If k_0 is exactly an integer and $\mathsf{k}^{(2)}$ converges to this limit by upper values, one would get

$$\lceil \mathsf{k}^{(2)} \rceil \underset{p \to 1}{\longrightarrow} \mathsf{k}_0 + 1. \tag{34}$$

However, it causes an error of one iteration in the worst case on k_{MG}, which is negligible for n large enough. Thus, we claim that one can take the configuration

$$(k_{\text{MG}}, T_{\text{MG}}) = \left(\lceil \mathsf{k}^{(0)} \rceil, \lceil T(\mathsf{k}_0, p) \rceil \right) \tag{35}$$

for p high enough, which is exactly the first configuration $(\lceil \mathsf{k}^{(1)} \rceil, \lceil \mathsf{T}^{(1)} \rceil)$ we encountered after solving the continuous case. It is easier to obtain since we have direct expressions to compute it in (20). We reported on Fig. 9 the behavior

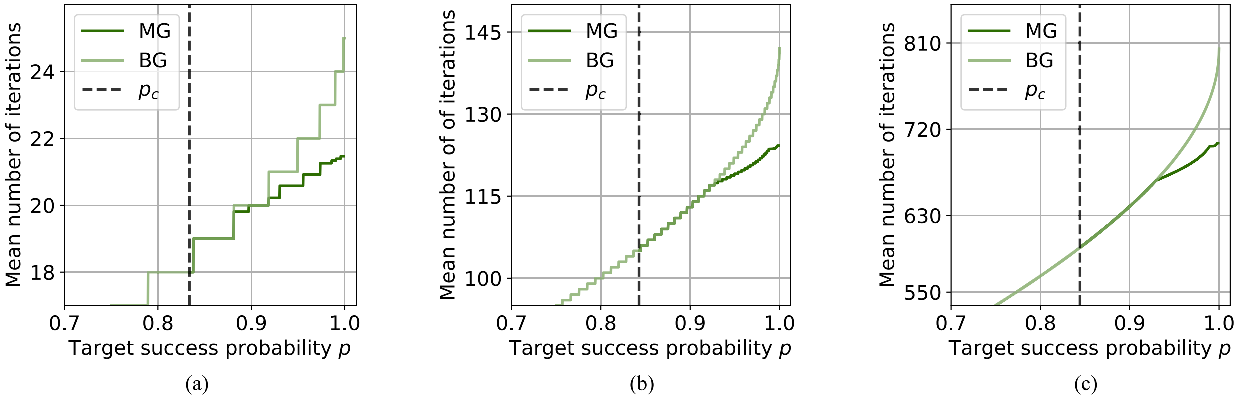


FIGURE 10. Expected number of iterations needed by MG and BG to reach a given success probability p at fixed number of users. We indicate p_c with the black dashed line. (a) n = 10 ($L_{\text{opt}} = 25$). (b) n = 15 ($L_{\text{opt}} = 142$). (c) n = 20 ($L_{\text{opt}} = 804$).

of $\lceil \mathsf{k}^{(2)} \rceil$ against p for $n = 10$ qubits to emphasize that the convergence is slow. We can see that replacing $\lceil \mathsf{k}^{(2)} \rceil$ by $\lceil \mathsf{k}_0 \rceil$ is justified for $p > 0.99$ in this example. Thus, we still advise to use the configuration of (32) if p is not very close to 1.

The vertical solid black line delimits the region where p is too close to 1 to be reached by BG. Indeed, the truncation with the upper integer part might result in

$$P_{\text{BG}}(L_{\text{opt}}) \lesssim 1. \tag{36}$$

In this example, where $n = 10$ qubits and $L_{\text{opt}} = 25$, the effective success probability of BG is $P_{\text{BG}}(L_{\text{opt}}) = 0.9995$. It is why L_{opt} is no longer indicated above $1 - p \le 10^{-4}$ on Fig. 9. On the other hand, MG has no such limitation since one can take T as large as necessary to reach a certain success probability (see Proposition 1).

VI. PERFORMANCE ANALYSIS

In this section, we compare the performance of our proposed MG algorithm with the BG ones.

A. GAIN OF ITERATIONS AT FIXED NUMBER OF USERS

Fig. 10 presents the expected number of iterations E_i obtained with our configuration $(k_{\text{MG}}, T_{\text{MG}})$ for a given p and compared to the performance of BG. As expected, the required number of iterations increases with the target success probability for both MG and BG. We can distinguish the following three parts in the plots.

1) When $p < p_c$, BG and MG curves are necessarily identical as we have shown that MG cannot bring improvement in this area.
2) There remains an uncertain region for $p \ge p_c$ where MG may or may not offer an advantage over BG. Our initial theoretical approach showed that the continuous version of the optimization procedure admits a solution with $\mathsf{T}^{(1)} > 1$ but it does not guarantee an integer pair $(k_{\text{MG}}, T_{\text{MG}})$ with $T_{\text{MG}} \ge 2$.
3) Finally as p increases, MG outperforms BG.

It seems that as n increases, it might be possible to define properly a boundary where MG becomes more interesting with certainty. However, the transition is not well defined in every case, as it can be seen with $n = 10$ and $n = 15$ qubits.

Nonetheless, MG requires always fewer (or the same) number of iterations than BG for the same target success probability.

B. VARYING THE NUMBER OF USERS

We can go further and visualize in the plane (p, n) the relative error

$$\frac{E_i^{\text{BG}} - E_i^{\text{MG}}}{E_i^{\text{BG}}} \tag{37}$$

in order to evaluate the gain of our algorithm as the number of qubits increases. E_i^{MG} corresponds to the expected number of iterations in the configuration of (32) and E_i^{BG} is the minimum number of iterations within Grover's algorithm to reach the success probability p

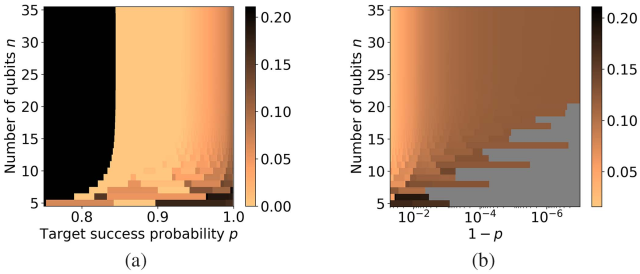
$$E_i^{\text{BG}}(p, n) = \left\lceil \frac{1}{2}\left(\frac{1}{\theta_G}\arcsin(\sqrt{p}) - 1\right)\right\rceil. \tag{38}$$

One recovers L_{opt} by taking the limit $E_i^{\text{BG}}(p \to 1, n)$.[1]

As mentioned before, the effective success probability of BG is not exactly 1 because of the truncation to an integer value to build L_{opt}. We take into account this effect by considering that E_i^{BG} is not defined if a further check reveals that $P_{\text{BG}}(E_i^{\text{BG}}(p, n)) < p$.

The results on Fig. 11(a)—where the black zone still correspond to $p < p_c$—suggest that our approach allows to reduce significantly the number iterations when p increases. Thus, we also show on Fig. 11(b) a zoom on the region where p is close to 1. The gray zone on the latter corresponds to the region where E_i^{BG} is not defined. This plot reveals that the ratio between the complexity of BG and MG seems to become independent of n as the number of qubits increases. This

[1] Because of the upper integer part $\lceil \cdot \rceil$, it might be $L_{\text{opt}} + 1$ but it is not significant as n increases.



(a)

(b)

FIGURE 11. **Relative error $(E_i^{\mathrm{BG}} - E_i^{\mathrm{MG}})/E_i^{\mathrm{BG}}$ between MG in our configuration and BG. (a) Full plane (p, n). (b) Zoom on $p \to 1$.**

emerging pattern allows us to state that our algorithm permits to reduce the complexity of Grover's search for $n \geq 15$ by 2% for $p \geq 0.95$, 6% for $p \geq 0.99$ and 10% for $p \geq 0.999$.

VII. CONCLUSION

In this work, we proposed to use Grover's algorithm in an hybrid format, which consists in performing the usual routine several times but with a reduced number of iterations.

Our main contribution is the theoretical derivation of the initial configuration for MG $(k_{\mathrm{MG}}, T_{\mathrm{MG}}) = (\lceil \mathsf{k}^{(2)} \rceil, T^{(2)})$ corresponding, respectively, to the number of iterations within one trial and the maximum number of trials. Given a target success probability p, we showed that our method ensures to find the marked element at most with the same complexity than Grover's algorithm. Indeed, for low p, our configuration corresponds exactly to the latter, i.e., $T_{\mathrm{MG}} = 1$. But, as p increases, MG definitely presents an advantage in terms of complexity. For instance, it allows to reduce by 10% the mean number of iterations for $p \geq 0.999$. Besides, it also permits to reach any success probability, contrarily to Grover.

One may note that this hybrid algorithm can be used instead of Grover's algorithm in any more complex algorithm relying on the latter.

It might be interesting to conduct further investigations on MG in a noisy environment. Indeed, it is easy to implement the model of the total depolarizing channel [21] in our equations by modifying the expression of P_{BG}. Then, an interesting improvement of this work could consist in analyzing how $(k_{\mathrm{MG}}, T_{\mathrm{MG}})$ would be modified against a noise parameter.

ACKNOWLEDGMENT

The authors thank Jean-Marie Gorce and Maxime Guillaud for their valuable discussions. The authors also thank the anonymous reviewer for their insightful comments and suggestions, which have helped improve the quality of this manuscript.

APPENDIX A
PROOF OF THEOREM 1

Let us prove the two points separately.

1) The equation $P_{\mathrm{MG}}(k, T) = p$ admits a unique solution at fixed k. It is obtained simply by inverting P_{MG}, which gives

$$T(k, p) = \frac{\log(1-p)}{\log(1 - P_{\mathrm{BG}}(k))}. \tag{39}$$

Recall also that for a given a pair (k, T) in $I_p^{(\mathbb{R}\times\mathbb{R})}$, one necessarily has $T \geq 1$. Hence, the set $I_p^{(1)} = I_p^{(\mathbb{R}\times\mathbb{R})} \bigcap \{(k, T)\,,\ P_{\mathrm{MG}}(k, T) = p\}$ reads

$$I_p^{(1)} = \{(k, T(k, p))\ ,\ \ 1 \leq k \leq L_{\mathrm{opt}}\ ,\ \ T(k, p) \geq 1\}. \tag{40}$$

2) Let us take (k, T) in $I_p^{(\mathbb{R}\times\mathbb{R})}$. Then, $P_{\mathrm{MG}}(k, T) \geq p$. Hence from (11) we have

$$E_i(k, T) = \frac{k}{P_{\mathrm{BG}}(k)} P_{\mathrm{MG}}(k, T) \geq \frac{k}{P_{\mathrm{BG}}(k)} \cdot p \tag{41}$$

which means $E_i(k, T) \geq E_i(k, T(k, p))$. Consequently, a minimizer of E_i over $I_p^{(\mathbb{R}\times\mathbb{R})}$ is necessarily located on $I_p^{(1)}$ and takes the form $(k, T) = (k, T(k, p))$.

Furthermore, $T(k, p) \geq 1$ if and only if $P_{\mathrm{BG}}(k) \leq p$. On the other hand, one has $E_i(k, T(k, p)) = f(k) \cdot p$ where we use the short-hand notation $f(k) = k/P_{\mathrm{BG}}(k)$. Consequently, k must minimize f under the constraint $P_{\mathrm{BG}}(k) \leq p$.

f admits a unique minimizer k_0 in $[1, L_{\mathrm{opt}}]$. It can be found by solving $\partial_k f = 0$, which gives exactly (21). Thus, we introduce the critical probability

$$p_c = P_{\mathrm{BG}}(\mathsf{k}_0) = \frac{16\mathsf{k}_0^2\theta_G^2}{1 + 16\mathsf{k}_0^2\theta_G^2} \tag{42}$$

which permits to distinguish the following two cases.

- *Case $p > p_c$:* Then, $P_{\mathrm{BG}}(\mathsf{k}_0) < p$, which implies that $k = \mathsf{k}_0$. Reciprocally, the pair $(\mathsf{k}_0, T(\mathsf{k}_0, p))$ is contained in $I_p^{(1)}$ and minimizes E_i on $I_p^{(\mathbb{R}\times\mathbb{R})}$. It ensures that
$$(\mathsf{k}^{(1)}, \mathsf{T}^{(1)}) = \left(\mathsf{k}_0, \frac{\log(1-p)}{\log(1-p_c)}\right). \tag{43}$$
- *Case $p \leq p_c$:* k_0 is too large to satisfy $P_{\mathrm{BG}}(\mathsf{k}_0) \leq p$, but f is decreasing toward its minimum on $[1, \mathsf{k}_0]$. Necessarily, k takes the largest allowed value corresponding to $P_{\mathrm{BG}}(k) = p$, in which case $T(k, p) = 1$. Reciprocally, this pair is indeed the solution which yields
$$(\mathsf{k}^{(1)}, \mathsf{T}^{(1)}) = \left(\frac{1}{2}\left(\frac{1}{\theta_G}\arcsin(\sqrt{p}) - 1\right), 1\right). \tag{44}$$

APPENDIX B
TECHNICALITIES ON THE DISCRETIZATION OF THE NUMBER OF TRIALS

In the main text, we defined the sequence

$$u[j] = \frac{1}{2}\left(\frac{1}{\theta_G}\arccos\left((1-p)^{1/2j}\right) - 1\right) \tag{45}$$

for $j \geq 1$, which is obtained by inverting the equation $T(u[j], p) = j$. It obeys the two following properties.

1) $(u[j])_{j\geq 1}$ decreases toward a limit $l < 0$.
2) For $j \geq 2$, one has

$$k \in [u[j], u[j-1]) \Longleftrightarrow \lceil T(k, p) \rceil = j.$$

Let us prove each point separately.

1) For $j \geq 1$, $(1-p)^{1/2j} \in (0, 1)$, thus the function $j \longrightarrow u[j]$ is decreasing on $\mathbb{R}_+^*$. Furthermore, using that $(1-p)^{1/2j}$ converges to 1, it is easy to get

$$u[j] \underset{j\infty}{\longrightarrow} -\frac{1}{2}. \tag{46}$$

2) By definition of $u[j]$, one has $T(u[j], p) = j$. Since $T(k, p)$ is a strictly decreasing function of k on $[1, L_{\text{opt}}]$, it ensures that

$$\lceil T(k, p) \rceil = j \;\; \forall k \in [u[j], u[j-1]). \tag{47}$$

The first point allows to safely define the integer m we use in the main text as

$$m = \min_{j\in\mathbb{N}^*} \{u[j] \leq \mathsf{k}_0\} \tag{48}$$

which allow to label the branch of $I_p^{(2)}$ on which the pair $(\mathsf{k}^{(2)}, T^{(2)})$ is located. In addition, the second point ensures that for $k \in [u[m], u[m-1])$, the associated (k, T) in $I_p^{(2)}$ verifies $T = \lceil T(k, p) \rceil = m$.

APPENDIX C
PROOF OF THEOREM 2

Since E_i is a growing function of T, one has for $k \in [1, L_{\text{opt}}]$

$$E_i(k, \lceil T(k, p) \rceil) \geq E_i(k, T(k, p)) \tag{49}$$

which is a formal way to say that the curve $E_i|_{I_p^{(1)}}$ is always below the curve of $E_i|_{I_p^{(2)}}$.

We are in the case $p > p_c$; hence the expected number of iterations along $I_p^{(1)}$ given by $E_i(k, T(k, p)) = f(k) \cdot p$ is decreasing for $k \in [1, \mathsf{k}_0]$ and increasing for $k \in [\mathsf{k}_0, L_{\text{opt}}]$. Furthermore

$$\begin{cases} u[m] \in [1, \mathsf{k}_0] & \text{and } T(u[m], p) = m \\ u[m-1] \in [\mathsf{k}_0, L_{\text{opt}}] & \text{and } T(u[m-1], p) = m-1 \end{cases} \tag{50}$$

which yields

$$\begin{cases} E_i(k, T(k, p)) > E_i(u[m], m) & \text{for } k < u[m] \\ E_i(k, T(k, p)) > E_i(u[m-1], m-1) & \text{for } k > u[m-1]. \end{cases} \tag{51}$$

Using (49), one can deduce that

$$\begin{cases} E_i(k, \lceil T(k, p) \rceil) > E_i(u[m], m) & \text{for } k < u[m] \\ E_i(k, \lceil T(k, p))\rceil > E_i(u[m-1], m-1) & \text{for } k > u[m-1]. \end{cases} \tag{52}$$

Finally, $(u[m], m)$ and $(u[m-1], m-1)$ are also contained in $I_p^{(2)}$. Notice that

1) E_i is continuous on the portion of $I_p^{(2)}$ where $u[m] \leq k < u[m-1]$;
2) $\lim\limits_{k\to u[m-1]} E_i(k, \lceil T(k, p) \rceil) \geq E_i(u[m-1], m-1)$.

Thus, E_i admits a minimizer $(\mathsf{k}^{(2)}, T^{(2)})$ either with $u[m] \leq k < u[m-1]$ or precisely at $(u[m-1], m-1)$. It corresponds as expected to

$$\begin{cases} u[m] \leq \mathsf{k}^{(2)} < u[m-1] & T^{(2)} = m \\ \text{or} & \\ \mathsf{k}^{(2)} = u[m-1] & T^{(2)} = m-1. \end{cases} \tag{53}$$

In practice, we compute $\mathsf{k}^{(2)}(p)$ by checking whether the derivative $\partial_k(E_i|_{I_p^{(2)}})$ [2] cancels on the branch of interest, and we compare the eventual solutions with the two boundary points $(u[m], m)$ and $(u[m-1], m-1)$. Such points, if they exist, are solutions in $[u[m], u[m-1])$ of the equation

$$\tan((2k+1)\theta_G) = 4k\theta_G - 4k\theta_G \frac{mP_{\text{BG}}(k)(1-P_{\text{BG}}(k))^{m-1}}{P_{\text{MG}}(k, m)}. \tag{54}$$

Notice that one recognizes (21) with a corrective term on the right-hand side. At the end one keeps the pair (k, T) with the lowest E_i.

APPENDIX D
PROOF OF THEOREM 3

By definition of $\mathsf{k}^{(2)}$, one has

$$u[m] \leq \mathsf{k}^{(2)} \leq u[m-1]. \tag{55}$$

Since $m = \lceil T(\mathsf{k}_0, p) \rceil$, one can write $m = T(\mathsf{k}_0, p) + y(p)$ with $y(p) \in [0, 1]$. Consequently

$$(1-p)^{\frac{1}{m}} = \exp\left(\log(1-p)\frac{\log(1-p_c)}{\log(1-p) + y(p)}\right) \underset{p\to 1}{\longrightarrow} 1 - p_c \tag{56}$$

since $y(p)$ is bounded. It yields

$$u[m] \underset{p\to 1}{\longrightarrow} \frac{1}{2}\left(\frac{1}{\theta_G}\arccos\left(\sqrt{1-p_c}\right) - 1\right) \tag{57}$$

and using the continuity $P_{\text{BG}}(k) = 1 - \cos^2((2k+1)\theta_G)$ one gets

$$P_{\text{BG}}(u[m]) \underset{p\to 1}{\longrightarrow} p_c = P_{\text{BG}}(\mathsf{k}_0). \tag{58}$$

Necessarily, $u[m]$ converges to k_0 as $p \to 1$ and the same conclusion holds for $u[m-1]$. Equation 55 allows to conclude

$$\mathsf{k}^{(2)} \underset{p\to 1}{\longrightarrow} \mathsf{k}_0. \tag{59}$$

[2] We use the notation ∂_k since we work on the branch where T is fixed to $\lceil T(k, p) \rceil = m$

REFERENCES

[1] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, Nov. 1996, pp. 212–219, doi: 10.1145/237814.237866.

[2] M. Boyer, G. Brassard, P. Hoeyer, and A. Tapp, "Tight bounds on quantum searching," *Fortschritte der Physik*, vol. 46, no. 4-5, pp. 493–505, Jun. 1998, doi: 10.1002/3527603093.ch10.

[3] C. Zalka, "Grover's quantum searching algorithm is optimal," *Phys. Rev. A*, vol. 60, no. 4, pp. 2746–2751, Oct. 1999, doi: 10.1103/PhysRevA.60.2746.

[4] C. Durr and P. Hoyer, "A quantum algorithm for finding the minimum," Jan. 1999, *arXiv:quant-ph/9607014*, doi: 10.48550/arXiv.quant-ph/9607014.

[5] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," Nov. 2014, *arXiv:1411.4028*, doi: 10.48550/arXiv.1411.4028.

[6] A. Peruzzo et al., "A variational eigenvalue solver on a quantum processor," *Nature Commun.*, vol. 5, no. 1, Jul. 2014, Art. no. 4213, doi: 10.1038/ncomms5213.

[7] A. Gilliam, S. Woerner, and C. Gonciulea, "Grover adaptive search for constrained polynomial binary optimization," *Quantum*, vol. 5, Apr. 2021, Art. no. 428, doi: 0.22331/q-2021-04-08-428.

[8] S. Mukherjee, "A Grover search-based algorithm for the list coloring problem," *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 3101008, doi: 10.1109/TQE.2022.3151137.

[9] A. Saha, D. Saha, and A. Chakrabarti, "Circuit design for K-coloring problem and its implementation on near-term quantum devices," in *Proc. IEEE Int. Symp. Smart Electron. Syst.*. Chennai, India, Dec. 2020, pp. 17–22, doi: 10.1109/iSES50453.2020.00015.

[10] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing Grover oracles for quantum key search on AES and LowMC," in *Proc. 39th Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, Oct. 2019, pp. 280–310, doi: 10.1007/978-3-030-45724-2_10.

[11] H. Tezuka, K. Nakaji, T. Satoh, and N. Yamamoto, "Grover search revisited; application to image pattern matching," *Phys. Rev. A*, vol. 105, no. 3, Mar. 2022, Art. no. 032440, doi: 10.1103/PhysRevA.105.032440.

[12] P. Botsinis et al., "Quantum search algorithms for wireless communications," *IEEE Commun. Surv. Tut.*, vol. 21, no. 2, pp. 1209–1242, 2019, doi: 10.1109/COMST.2018.2882385.

[13] E. M. Stoudenmire and X. Waintal, "Opening the black box inside Grover's algorithm opening the black box inside Grover's algorithm," *Phys. Rev. X*, vol. 14, no. 4, Nov. 2024, Art. no. 041029, doi: 10.1109/COMST.2018.2882385.

[14] M. I. Habibie, C. Goursaud, and J. Hamie, "Quantum minimum searching algorithms for active user detection in wireless IoT networks," *IEEE Internet of Things J.*, vol. 11, no. 12, pp. 22603–22615, Jun. 2024, doi: 10.1109/JIOT.2024.3382337.

[15] H. Liu, F. Li, and Y. Fan, "Optimizing the quantum circuit for solving Boolean equations based on Grover search algorithm," *Electronics*, vol. 11, no. 15, Jan. 2022, Art. no. 2467, doi: 10.3390/electronics11152467.

[16] X. Wu et al., "Circuit optimization of Grover quantum search algorithm," *Quantum Inf. Process.*, vol. 22, no. 1, Jan. 2023, doi: 10.1007/s11128-022-03727-y.

[17] L. Cherckesova, O. Safaryan, I. Pilipenko, V. Porksheyan, N. Bogdanova, and N. Beryoza, "Modification of the quantum Grover algorithm by using the inversion method around the middle," *IOP Conf. Series: Mater. Sci. Eng.*, vol. 1001, no. 1, Dec. 2020, Art. no. 012065, doi: 10.1088/1757-899X/1001/1/012065.

[18] M. Kessler, D. Alonso, and P. Sánchez, "Determination of the number of shots for Grover's search algorithm," *EPJ Quantum Technol.*, vol. 10, no. 1, Dec. 2023, Art. no. 47, doi: 10.1140/epjqt/s40507-023-00204-y.

[19] A. Mandviwalla, K. Ohshiro, and B. Ji, "Implementing Grover's algorithm on the IBM quantum computers," in *Proc. 2018 IEEE Int. Conf. Big Data*, Dec. 2018, pp. 2531–2537, doi: 10.1109/BigData.2018.8622457.

[20] Y. E. Kaderi, A. Honecker, and I. Andriyanova, "Performance of uncoded implementation of Grover's algorithm on today's quantum processors," in *Proc. 2023 IEEE Inf. Theory Workshop*, Apr. 2023, pp. 209–214, doi: 10.48550/arXiv.2212.10482.

[21] I. Cohn, A. L. F. de Oliveira, E. Buksman, and J. G. L. de Lacalle, "Grover's search with local and total depolarizing channel errors," *Int. J. Quantum Inf.*, vol. 14, no. 02, Mar. 2016, Art. no. 1650009, doi: 10.1142/S021974991650009X.

[22] M. I. Habibie, J. Hamie, and C. Goursaud, "A performance comparison of classical and quantum algorithm for active user detection," in *Proc. 23rd IEEE Int. Workshop Signal Process. Adv. Wirel. Commun.*, Oulu, Finland, Jul. 2022, pp. 1–5, doi: 10.1109/SPAWC51304.2022.9833942.

[23] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, and C. Monroe, "Complete 3-Qubit Grover search on a programmable quantum computer," *Nature Commun.*, vol. 8, no. 1, Dec. 2017, Art. no. 1918, doi: 10.48550/arXiv.1703.10535.

[24] J. Zhu, Y. Gao, H. Wang, T. Li, and H. Wu, "A realizable GAS-based quantum algorithm for traveling salesman problem" 2022, *arXiv:2212.02735*, doi: 10.48550/arXiv.2212.02735.

[25] E. Biham, O. Biham, D. Biron, M. Grassl, and D. A. Lidar, "Grover's quantum search algorithm for an arbitrary initial amplitude distribution," *Phys. Rev. A*, vol. 60, no. 4, pp. 2742–2745, Oct. 1999, doi: 10.48550/arXiv.quant-ph/9807027.

[26] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signal Process.*, vol. 11, no. 3-4, pp. 154–655, 2017, doi: 10.1561/2000000093.

Romain Piron (Student Member, IEEE) received the M.Sc. degree in engineering from CentraleSupélec, Gif-sur-Yvette, France, and the M.Sc. degree in physics from EPFL, Lausanne, Switzerland, in 2023. He is currently working toward the Ph.D. degree with INSA Lyon, Villeurbanne, France, working on the adaptation of quantum optimization algorithms for activity detection in massive wireless networks, under the supervision of Claire Goursaud.

He has presented his work in international conferences. His research focuses on quantum algorithms, with a particular emphasis on quantum annealing for combinatorial optimization in signal processing applications.

Muhammad Idham Habibie (Student Member, IEEE) received the Bachelor's degree in electrical and electronics engineering with University of Indonesia, Depok, Indonesia, in 2013, M.Sc. degree in telecommunications engineerin with Cobham Wireless from the University College London, London, U.K, in 2017, and the Ph.D. degree in signal processing from the Institut National des Sciences Appliquées de Lyon, Villeurbanne, France under the supervision of Claire Goursaud, in 2023.

He has authored or coauthored several articles related to quantum communication, Grover's algorithm. His research interests include quantum computation, quantum algorithms, and quantum active user detection.

Claire Goursaud (Member, IEEE) received the Ph.D. degree in high frequency and optical telecommunications from the University of Limoges, Limoges, France, in 2006, working on signal processing for optical communications.

In September 2007, she joined the Institut National des Sciences Appliquées de Lyon as an Assistant Professor with the Telecommunication Department and the CITI Laboratory. She has authored or coauthored more than 60 refereed journal and conference papers Her research interests include massive multiple access for wireless networks, and particularly the use of quantum algorithms and SNN architecture to tackle the signal processing challenge.

Dr. Goursaud is an Associate Editor for two international journals.