Engineering Applications of Artificial Intelligence 121 (2023) 106058

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Quantum hybrid algorithm for solving SAT problem

Charles Moudina Varmantchaonala [a], Jean Louis Kedieng Ebongue Fendji [b,c,*], Jean Pierre Tchapet Njafa [d], Marcellin Atemkeng [e]

[a] Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundere, Ngaoundere, Po. Box 454, Cameroon
[b] Department of Computer Engineering, University Institute of Technology, University of Ngaoundere, Ngaoundere, Po. Box 455, Cameroon
[c] Department of Fundamental Science and Engineering Techniques, National School of Chemical Engineering and Mineral Industries, University of Ngaoundere, Po. Box 454, Cameroon
[d] Department of Physics, Faculty of Science, University of Maroua, Maroua, Po. Box 46, Cameroon
[e] Department of Mathematics, Rhodes University, Grahamstown, Po. Box 6140, South Africa

ARTICLE INFO

Keywords:
Quantum algorithm
Grover algorithm
Combinatorial problem
SAT problem

ABSTRACT

Combinatorial problems usually have a large search space, and almost all classical algorithms for solving this class of problems are inefficient for real-life input sizes. Quantum algorithms have been introduced with the aim of reducing computational time. In addition, to speed up the computation while solving combinatorial problems, another approach is to reduce the search space and focus on a restricted area. In this work, we study a parameter that helps reduce the search space for the solution to the satisfiability problem (SAT). We propose an approach based on Grover's algorithm that considers the defined parameter to reduce the search space for the solution of the SAT problem. Our algorithm has a complexity of the order $\mathcal{O}(\sqrt{N/S})$, or $\mathcal{O}(\sqrt{N/lS})$ if there are l good solutions among the initial solutions, with N the number of potential solutions and S the reduction or subdivision factor of the space.

1. Introduction

Quantum computing consists of developing algorithms inspired by the properties of quantum physics such as superposition and entanglement. This new generation of algorithms is capable of solving most difficult problems for classical supercomputers with great efficiency. Among these difficult problems, there are combinatorial optimization problems.

Combinatorial optimization problems, especially the satisfiability (SAT) problem, find their applications in several areas of life such as social networks (Jabbour et al., 2020), the design of computer systems (Nam et al., 2018; Bryant and Heule, 2021; He et al., 2018; Khurshid and Marinov, 2004; Tucker et al., 2007), planning (Gasquet et al., 2018; Froleyks et al., 2021; Erós et al., 2021), pedigree consistency checking (Konovalenko, 2019), generation of test pattern used for digital systems (Stava, 2021), design debugging and diagnosis (Gaber et al., 2020), integrated circuit design (McGeer et al., 1991), identification of functional dependencies in boolean functions (Dixit and Kolaitis, 2019), diabetes detection and bioinformatic (Kasihmuddin et al., 2018; Luo et al., 2022), just to name a few.

Solving combinatorial problems requires a factorial time, which remains a challenge on classical computers, even for small real-life input sizes, due to the search space of solutions, which in general is very large. Most of the existing classical algorithms are known to have factorial complexity, and therefore they are inefficient. This limitation has fostered the development of several quantum algorithms for combinatorial optimization problems.

One of the largely used algorithms that has been the basis of several others is Grover's algorithm. It allows us to find a good solution with a complexity of $\mathcal{O}(\sqrt{N})$ against $\mathcal{O}(N)$ classically, with N the total number of elements or potential solutions. Several Grover-based quantum algorithms have been developed to solve combinatorial problems, including the 3 -SAT problem (Yang et al., 2009; Cheng and Tao, 2007; Leporati and Felloni, 2007; Zhang et al., 2020; Campos et al., 2021; Fernandes and Dutra, 2019). But most of these algorithms based on Grover start with a uniform superposition of all the potential solutions or elements and work on the principle of amplification, which consists of augmenting the amplitude of the good solution compared to the others.

The fact that there is equiprobability or a uniform superposition requires many iterations before finding the good solution, and the quantum state thus obtained by superposition must follow a law of unitarity of the norm. However, the computation time can be reduced

* Corresponding author at: Department of Fundamental Science and Engineering Techniques, National School of Chemical Engineering and Mineral Industries, University of Ngaoundere, Po. Box 454, Cameroon.

E-mail addresses: var001charmant@gmail.com (C.M. Varmantchaonala), jlfendji@univ-ndere.cm (J.L.K.E. Fendji), jean-pierre.tchapet-njafa@univ-maroua.cm (J.P.T. Njafa), m.atemkeng@ru.ac.za (M. Atemkeng).

https://doi.org/10.1016/j.engappai.2023.106058
Received 26 May 2022; Received in revised form 1 February 2023; Accepted 23 February 2023
Available online 2 March 2023
0952-1976/

by searching for the right search space of the solution. In fact, a good technique for solving combinatorial problems is the preliminary study of parameters that can reduce the search space. Such a reduction of the search space can disturb the uniformity of the initial superposition, and consequently speed up the computation. Thus, the objective of this work is to decrease the computation time of algorithms for combinatorial optimization problems by introducing a disturbance of the initial uniform superposition. By applying the same technique, a data parallelism approach in Grover's algorithm is described.

The remainder of this paper is organized as follows. Section 2 presents the basics of quantum computing. Section 3 summarizes the related work and our contributions. Section 4 presents the SAT problem and in Section 5 introduces a factor that can influence the solution of the SAT problem. Section 6 presents the Grover algorithm and Section 7 presents the proposed algorithm and studies its complexity. The Section 8 is dedicated to the results and discussions, and finally Section 9 concludes the article and provides future research directions.

2. Quantum computation

The smallest unit of information in quantum computing is called the qubit. To represent the behavior of quantum phenomena, Dirac introduced an adapted notation (Munoz and Delgado, 2016) to describe the states of quantum systems and the transformations that act on them. This notation is called the bra/ket notation. The bit in classical computation takes two possible values, 0 and 1. In contrast, in quantum computing (QC), the quantum bit takes the value $|0\rangle$ (Dirac notation) for 0 and $|1\rangle$ for 1 and, moreover, another possible state constituted by the superposition of the two basic states $|0\rangle$ and $|1\rangle$.

Therefore, in general, a qubit can be represented in the superposition state by Eq. (1) or Eq. (2).

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \tag{2}$$

where α and β are complex numbers that specify the probability amplitudes of the corresponding states, with the condition:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3}$$

We can return to the basic states $|0\rangle$ or $|1\rangle$ respectively if $\alpha = 0$ or $\beta = 0$. $|\alpha|^2$ and $|\beta|^2$ represent the probability to measure either $|0\rangle$ or $|1\rangle$ respectively. When measuring a qubit or a quantum state in general, this state reduces to the ground state in the superposition that has the highest probability.

If the state or quantum system consists of several (m qubits), we have the following.

$$\cdot\begin{bmatrix}\alpha_1 \mid & \alpha_2 \mid & \dots & \mid \alpha_m\\ \beta_1 \mid & \beta_2 \mid & \dots & \mid \beta_m\end{bmatrix} \tag{4}$$

According to Eq. (3), $|\alpha_i|^2 + |\beta_i|^2 = 1, i = 1, 2 \cdots, m$.

Thus, for instance, if there are three qubits in the system, we have the following.

$$\begin{bmatrix}\frac{1}{\sqrt{2}} \mid & \frac{1}{\sqrt{2} \mid} & \mid \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} \mid & \frac{1}{\sqrt{2} \mid} & \mid \frac{1}{\sqrt{2}}\end{bmatrix} \tag{5}$$

and, therefore, the final system is represented as:

$$|\psi\rangle = \frac{1}{\sqrt{2^3}}\Big[|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big] \tag{6}$$

The probability of finding the final system, Eqs. (6) and (5), during measurement in states $|000\rangle$, $|001\rangle$, $|100\rangle$, $|101\rangle$, $|010\rangle$, $|011\rangle$, $|110\rangle$, and $|111\rangle$ is $\frac{1}{8}$, for a uniform superposition.

The main advantage of the representation of information with qubits is the number of simultaneous values that can be carried by states or quantum systems. Thus, a state as Eq. (1) carries at the same time or simultaneously two basic values $|0\rangle$ and $|1\rangle$. Consequently, a quantum state made of n qubits represents 2^n information or values simultaneously (the quantum system in Eq. (6) or (5) represents 8 classical values at the same time). Moreover, in traditional or classical data processing, a sequence of n bits can represent only one of the 2^n possibilities.

3. Related works

The SAT problem has been widely discussed in the scientific literature. It is a combinatorial problem of great theoretical and practical interest. Like all combinatorial problems, finding a solution is a time-consuming operation, which increases factorially with the size of inputs. It is one of the first problems to be recognized as NP-Complete (Cook, 1971). Because of its practical interest, several methods have been proposed for its resolution, both classical and quantum based.

A k-SAT problem is informally a set of k variables and n clauses defined on these variables. Several classical algorithms have been proposed to efficiently solve instances of the SAT problem. Most of these algorithms are heuristics. The efficiency of these algorithms depends on the density $\alpha = n/m$ of the SAT formulas. For a SAT problem of density $\alpha \geq 2^k \log 2$, this problem is unsatisfiable with high probability (Franco and Paull, 1983) and conversely, for a density of $\alpha < 2^k/k$, the SAT problem is satisfiable with a high probability (Ming-Te and Franco, 1990; Chvátal and Reed, 1992). Table 1 shows the different densities of the clauses and the algorithms that can solve them efficiently.

More recently (Ishtaiwi et al., 2021) proposed an algorithm based on the dynamic technique, which consists in assigning weights to clauses. It subdivides the clauses into four categories: (1) small-size clauses linked with a small neighborhood, (2) small-size clauses and linked with a large neighborhood, (3) large-size clauses and linked with a small neighborhood, and (4) large-size clauses and linked with a large neighborhood. The conclusion of this method is that its performance varies only slightly with the size of the problem. Hansen et al. (2019) used the PPSZ bias to improve the PPZ algorithm (Paturi et al., 1997). The proposed method allows us to improve the complexity from 1.308^n to 1.307^n, but it is still a heuristic. Therefore, it is an approximation of the solution if there is one. The election Algorithm for Random k-Satisfiability in the Hopfield neural network has recently been proposed (Sathasivam et al., 2020). The election algorithm (EA) is a novel variant of the socio-political metaheuristic algorithm, inspired by the presidential election model conducted globally. It uses the Hopfield Neural Network to generate global solutions to the random problem of k -satisfiability.

Some classical methods, i.e. using classical computers, can solve this problem efficiently, but they are limited when the size of the inputs becomes important, as is the case with other combinatorial problems in general. This explains why in recent years, great interest has been focused on new techniques such as quantum computing. Quantum algorithms that have been developed to solve the SAT problem fall into two categories: metaheuristics and exact algorithms.

Metaheuristics are algorithms that generate approximative and good-quality solutions in a reasonable time. These approximate solutions may not be optimal, but rather close to optimal. From this category, we note the Quantum Adiabatic Algorithm, QAA (Farhi et al., 2001, 2002) and its applicable derivatives (Goto et al., 2019; Steffen et al., 2003; Kinjo et al., 2005; Lin et al., 2020) and the Quantum Simulated Annealing, QA (Apolloni et al., 1989; de Falco and Tamascelli, 2011) with its variants (Campos et al., 2021; Battaglia et al., 2005; King and McGeoch, 2014; Somma et al., 2008). We can also quote hybrid algorithms such as the quantum approximative optimization Algorithm, QAOA (Farhi et al., 2014; Farhi and Harrow, 2016; Guerreschi and Matsuura, 2019) with its improvements (Zhu et al., 2022; Egger et al., 2021; Khairy et al., 2020; Farhi et al.,

Table 1
Algorithmic hardness results for a random k-SAT with large k.

Reference	Algorithm or algorithm class	Clause density
Achlioptas et al. (2004)	DPLL algorithms	$O_k(2^k/k)$
Gamarnik and Sudan (2017)	Balanced sequential local algorithms (NAE-k-SAT)	$(1+O_k(1))2^{k-1}\log^2 k/k$
Hetterich (2016)	Survey Propagation guided decimation	$(1+O_k(1))2^k \log k/k$
Coja-Oghlan et al. (2017)	Walksat	$2^k \log^2 k/k$
Bresler and Huang (2022)	Low degree polynomials	$(1+O_k(1))k*2^k \log k/k$

2017). The principle in these approximate methods is the evolution of quantum systems from one energy level to another. In general, the combinatorial problem by its cost function is transformed into a Hamiltonian which itself is mapped to a quantum system. Most often, the idea is to make this system evolve towards a low-energy level, which constitutes the desired solution to the problem. In particular, the SAT problem is transformed into its variant, called MAX-SAT. The latter is seen as finding an assignment that violates fewer clauses of the SAT problem. This formulation is mapped into a quantum system. The Hamiltonian, which is the equivalent of the cost function of the problem, evolves towards the lowest energy level (the smallest number of violated clauses, which is the solution). But we recall once more that these solutions in general are approximate and may not be optimal.

The exact methods are those which try to find the optimal solution. This category of methods consists of doing an exhaustive search on the whole search space. One of the most common algorithms in this category remains Grover's algorithm (Grover, 1996; Biham et al., 1999). Grover's algorithm is a quantum algorithm that allows one to find a good element among N elements with a computation time of the order $\mathcal{O}(\sqrt{N})$ versus $\mathcal{O}(N)$ classically. During the last years, a variety of quantum algorithms have been developed, inspired by new quantum techniques and classical algorithms. Among these quantum algorithms for the SAT problem, we have quantum algorithms for the generic Boolean satisfiability problem (Yang et al., 2009; Wang et al., 2020). Many of these quantum algorithms are based on Grover, but they make a uniform superposition of the initial solutions with other techniques. Using Grover's search in a reduced space, Mandra et al. (2016) proposed an exact method for solving the exact SAT problem with a complexity of $\mathcal{O}(2^{n-M'})$, where n is the number of variables and M' the number of linearly independent clauses. The proposed method consists of identifying a subspace that contains valid solutions to the SAT problem and then searching in this restricted area. The process of reduction of the space used is the one that consists of reducing an SAT instance to its XORSAT equivalent that has a reduced solution space. Using a combination of quantum computation, Grover search and classical computation, Leporati and Felloni (2007) proposed a solution to the SAT problem, the Quantum Cooperative Search Algorithm (QCS), which consists of reducing the number of auxiliary qubits used by replacing them with classical ones. Chou et al. (2022) proposed a solution to the Max-k-XOR problem by linking it to mean-field p-spin glasses. Campos et al. (2021) uses quantum tunneling effects combined with quantum walks to solve hard K-SAT instances. The heuristic algorithm aims to iteratively reduce the Hamming distance between an evolving state $|\psi_j\rangle$ and a state that represents a satisfying assignment. One of the main problems of Grover's algorithm is the number of quantum circuits to be used to build the algorithm. Thus, to reduce this number, Wang et al. (2020) used circuit synthesis by designing a generic input for an instance of random 3-SAT.

Finding an exact solution to a problem is the most desirable case, even for large-scale combinatorial problems, since an approximate solution may make approximate and risky decisions. However, the exact methods may be time-consuming. Based on Grover's algorithm, we propose an exact algorithm for solving the SAT problem that performs an exhaustive search and considerably reduces the computation time. The idea is to avoid a uniform superposition in order to reduce the search space based on an influence factor of the SAT solution.

Our contributions are as follows.

- We propose a deterministic algorithm. The majority of the algorithms mentioned above are heuristics. The guarantee of the optimal solution is a probabilistic event. However, the proposed algorithm, using Grover's search, allows for an exhaustive search of all solutions to determine the optimal one.
- We introduce a new parameter that can influence the solution of the SAT problem. Parameters such as the density of clauses (Bresler and Huang, 2022) have been shown to be of high difficulty in solving the SAT problem. The positiveness/negativeness ratio introduced in this paper can be easily computed. To the best of our knowledge, this is the first quantum algorithm to use such a ratio to subdivide the search space. This is done in contrast to other techniques such as the conversion of the SAT problem into its equivalent XORSAT (Mandra et al., 2016), or the subdivision of the clauses into small and large size categories (Ishtaiwi et al., 2021).
- We improve the complexity with the new quantum algorithm proposed. Deterministic algorithms based on the Grover search algorithm generally have a complexity of $\mathcal{O}(\sqrt{N})$ but with the process of subdivision of the search space, we propose an algorithm based on Grover's search whose complexity is $\mathcal{O}(\sqrt{N/S})$, or $\mathcal{O}(\sqrt{N/lS})$ if there are l good solutions among the initial solutions, with N the number of potential solutions and S the reduction or subdivision factor of the space.

4. SAT problem

Let $x_1, x_2, \ldots, x_n$ be binary variables that take the value 0 or 1. And let $\neg$, $\vee$, and $\wedge$ be the logical `NOT` operator, the logical `OR` connector, and the logical `AND` connector, respectively.

Definition 1. A literal is a variable x or its negation $\neg x$. A clause is a disjunction of literals $C = (x_1 \vee \cdots \vee x_n)$, with each variable x_i, where $i = 1, \ldots, n$, which can be represented or replaced by its opposite $\neg x_i$ or may not exist.

Definition 2. A SAT problem or a Boolean satisfiability problem is defined by a formula F consisting of conjunctions of a set of binary variable clauses using logical operators `OR` ($\vee$) and `NOT` ($\neg$) in variables and `AND` ($\wedge$) between clauses as follows:

$$F(x_1, x_2, \ldots, x_n) = (x_1 \vee x_2 \vee \cdots \vee x_n) \wedge (x_1 \vee x_2 \vee \cdots \vee x_n) \wedge \cdots . \tag{7}$$

An example of a 3 -SAT formula is as follows:

$$F(x_1, x_2, x_3) = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_3). \tag{8}$$

A solution of an SAT problem is an assignment of binary variables that allows us to let $F(x_1, x_2, \ldots, x_n) = 1\,(true)$.

From our previous example, the assignment 111 is a solution, since

$$F(1,1,1) = (\neg 1 \vee 1 \vee 1) \wedge (1 \vee \neg 1 \vee \neg 1) \wedge (1 \vee 1 \vee 1) = 1\,(true). \tag{9}$$

In contrast, assignment 011 is not a solution, since

$$F(0,1,1) = (\neg 0 \vee 1 \vee 1) \wedge (0 \vee \neg 1 \vee \neg 1) \wedge (0 \vee 1 \vee 1) = 0\,(false). \tag{10}$$

The definition given here is valid for all k-SAT problems with the difference that a given k will indicate the maximum number of variables in the formula clauses.

Table 2
Search space.

x1	x2	x3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Definition 3. An XOR-clause $C = (x_1 \oplus \cdots \oplus x_n)$ is a set of literals that involve distinct variables. An XOR-formula is a conjunction of (not necessarily distinct) non-empty XOR-clauses. Assignment A satisfies an XOR-clause $C = (x_1 \oplus \cdots \oplus x_n)$ iff $A(C) = \sum_{i=1}^{n} A(x_i) mod 2 = 1$

Definition 4. A Horn clause is a clause consisting of a disjunction of literals, in which at most one variable is positive. A Horn-SAT formula is a conjunction of Horn clauses.

5. Study of the influence factor of the SAT solution

In this section, we first study a factor that affects the position of the solution of the SAT problem to reduce the search space. We are looking for a way to decrease or divide the search space that will guide the decrease in the amplitude of the potential bad solutions and the increase in the amplitude of the potential good ones.

The following observations are made:

- Case 1: All variables are positive.
  For an SAT problem where *all variables are positive*, that is, without $\neg x_i$, a solution is the assignment of 1 to all variables, that is, $1111111 \cdots 1$.
  As an example, for

$$F(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \quad (11)$$

  a solution is 111. The search space consisting of all potential solutions is given in Table 2.
  A solution of Case 1 is found at the end of the search space.
- Case 2: All variables are negative.
  The same way applies for an SAT problem where *all variables are negative*, a solution is to assign 0 to all variables, i.e., $0000000 \cdots 0$. As an example, for

$$F(x_1, x_2, x_3) = (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \quad (12)$$

  a solution is 000.
  A solution of Case 2 is at the beginning of the search space as shown in Table 2.
- Case 3: Attempt of generalization.
  This is the case where clauses contain both positive and negative variables.
  Tables 3 and 4 attempt to generalize the study to the following cases:
  - The majority of the variables are negative;
  - The majority of the variables are positive;
  - The case where the variables are balanced (balance between the positiveness and negativeness of the variables).

Conventions

1. Positiveness: is the number of positive variables in the SAT formula.
2. Negativeness: is the number of negative variables in the SAT formula.
3. Let us consider :
   - the end of the search space as $+\infty$;
   - the beginning of the search space as $-\infty$;
   - the middle of the search space as *mean*.

A SAT problem can have several solutions. Tables 3 and 4 estimate the number of solutions found in the search area ($+\infty$, $-\infty$ or the mean) in percentage that expresses the degree of determinism to find a solution in the area.

Discussion of the results of this observation

It appears from Tables 3 and 4 that the positiveness or negativeness of the variables is a deterministic factor in the position of the solution.

For the 3-SAT we have the following results:

Table 3
Study of 2-SAT: PS (Potential solutions), S (solution), S(∞): Solutions at ∞.

Search space	PS at $+\infty$	PS at the mean	PS at $-\infty$	S($+\infty$)	S($-\infty$)	% S($+\infty$)	% S(mean)	% S($-\infty$)
2-SAT								
$A = (x_1 \vee x_2) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_2)$								
00	11	01	00	11	01	100%	100%	50%
01	10	10	01	10				
10								
11								
$A = (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$								
00	11	01	00	10	00	50%	100%	100%
01	10	10	01		01			
10								
11								
$A = (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$								
00	11	01	00		00	0%	50%	100%
01	10	10	01		01			
10								
11								
$A = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2)$								
00	11	01	00	10	01	50%	100%	50%
01	10	10	01					
10								
11								

Table 4
Study of 3-SAT: PS (Potential solutions), S (solution), S(∞): Solutions at ∞.

Search space	PS at +∞	PS at the mean	PS at −∞	S(+∞)	S(−∞)	% S(+∞)	% S(mean)	% S(−∞)
3-SAT								
$A = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$								
000	111	101	000	111	001	100%	100%	66%
001	110	100	001	110	010			
010	101	011	010	101				
011								
100								
101								
110								
111								
$A = (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$								
000	111	101	000	110	000	66%	100%	100%
001	110	100	001	101	001			
010	101	011	010		010			
011								
100								
101								
110								
111								
$A = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$								
000	111	101	000	111	001	66%	100%	66%
001	110	100	001	101	010			
010	101	011	010					
011								
100								
101								
110								
111								

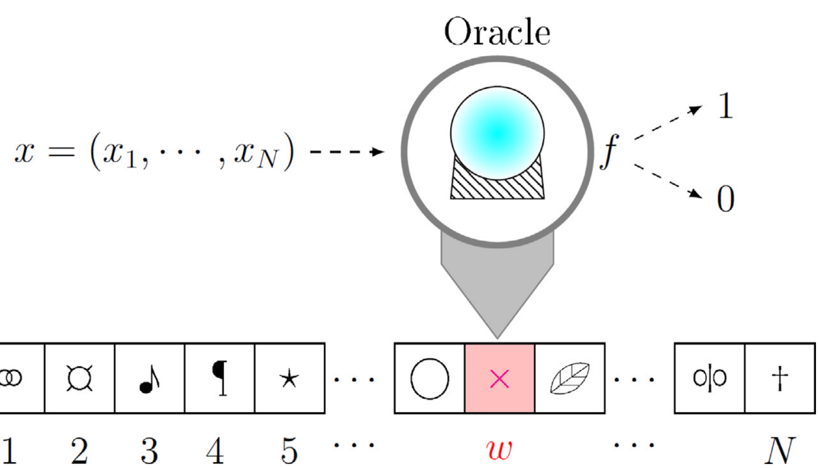

Fig. 1. Quantum oracle.

1. When all or most variables are positive, the degree of determinism, that is, the degree of finding a solution in the area +∞ is 100%.
2. When there is a balance between the negativeness and the positiveness of the variables, the degree of determinism of a solution in the mean area is 100%.
3. When all variables or most variables are negative, the degree of determinism of a solution in the area −∞ is 100%.

6. Grover algorithm

The Grover algorithm is a search algorithm that allows you to find one or more elements, for example w, that meet a given criterion in an unsorted list of N elements.

A typical example is the search in a database, or a telephone directory, of the name that corresponds to a given telephone number.

By exploiting quantum properties, the Grover algorithm uses a quantum oracle to achieve its goals. This oracle efficiently checks whether an element respects a given search criterion or not (see Fig. 1). For the resolution of the SAT problem, the oracle checks if an assignment leaves the SAT formula true or false.

To classically find the item w in the list by querying the classical oracle, that is, one that works according to the principle of our current computers, one needs at least $N/2$ queries or tests, and in the worst case N queries. Therefore, we say that the classical algorithm solves the problem in $\mathcal{O}(N)$.

On the other hand, the Grover quantum algorithm is an algorithm that finds an answer in $\mathcal{O}(\sqrt{N})$, even in the worst case, which is a quadratic acceleration compared to a classical naive algorithm.

Let us consider the function

$$f : (x_1, \ldots, x_n) \mapsto f(x_1, \ldots, x_n) \in \{0, 1\}. \tag{13}$$

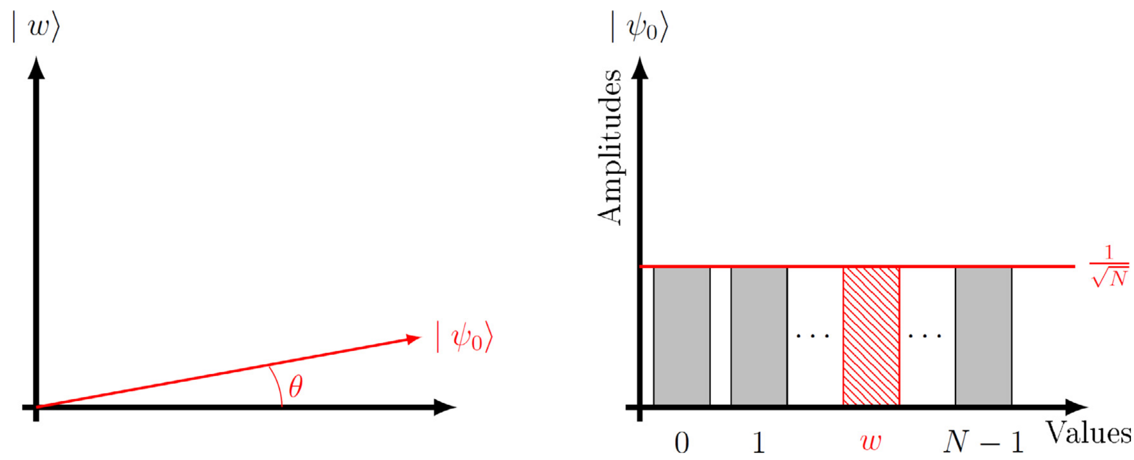

Fig. 2. Uniform superposition of potential solutions.

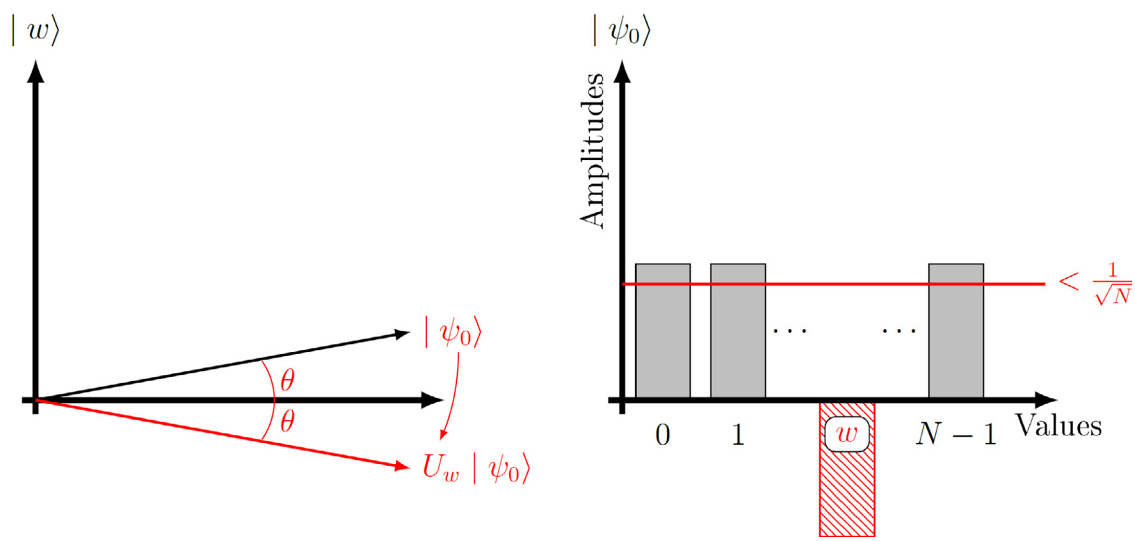

Fig. 3. Inversion of the solution.

The oracle is considered as a function, integrated in a box that we can still interrogate at any time. We can express the oracle as a unit operator U_w, defined by:

$$\begin{aligned} U_w|x\rangle|y\rangle &= |x\rangle|y \oplus f(x)\rangle \\ &= (-1)^{f(x)}|x\rangle. \end{aligned} \tag{14}$$

The quantum aspect of the algorithm is mainly in the superposition of states, in the form of a uniform state of amplitudes (created by applying the gates of Hadamard).

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |x\rangle. \tag{15}$$

Since the superposition is uniform, the probability of finding the correct value w is $1/N$ (corresponding to an amplitude of probability range of $1/\sqrt{N}$), and so it would take N trials on average. See as example Fig. 2. The angle θ on the left part of the figure is obtained according to the relation: $\sin\theta = 1/\sqrt{N}$.

Now, when we apply the oracle to our initial uniform superposition state $|\psi_0\rangle$, we have the following.

$$U_w|\psi_0\rangle = (-1)^{f(x)}|\psi_0\rangle. \tag{16}$$

This result is a mirror transformation with respect to w because we have:

$$U_w|w\rangle = I - 2|w\rangle\langle w|. \tag{17}$$

As a result, the amplitude corresponding to w is therefore negative, and it follows that the average of the amplitudes is decreased and thus ends less than $1/\sqrt{N}$ (see Fig. 3).

Then, it is necessary to apply the Grover operator:

$$G_{\psi_0} = 2|\psi_0\rangle\langle\psi_0| - I. \tag{18}$$

It is a process that allows one to highlight the solution indexed by the quantum oracle by performing a process called diffusion that consists of emphasizing or increasing the amplitude of this good solution. This rotation operation brings the current state closer to the solution

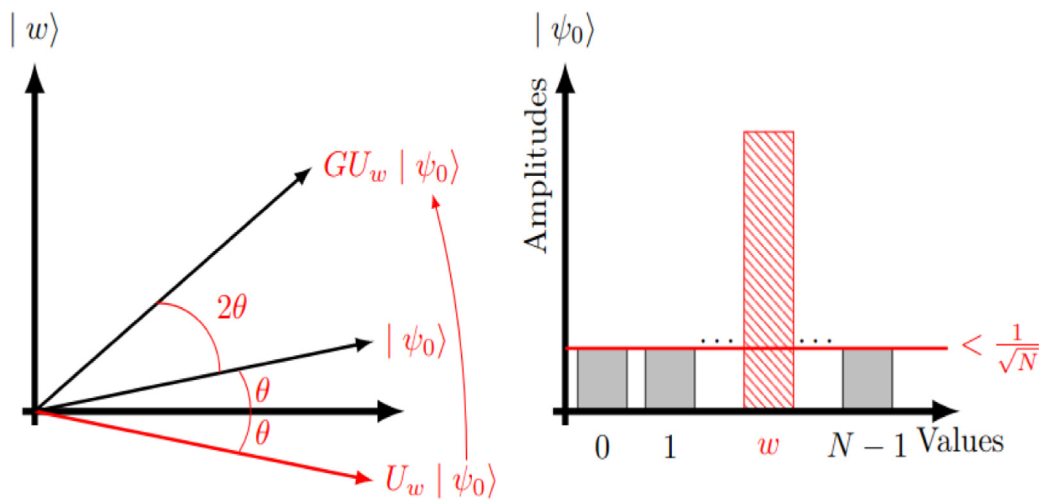

Fig. 4. Amplitude amplification of the good solution.

$|w\rangle$. The amplitude corresponding to $|w\rangle$ increases (about 3 times its original value, precisely in the case with four possible solutions), and the others decrease (see Fig. 4).

Thus, the Grover algorithm consists of repeating both operations several times: the oracle and the diffusion. This is done each time along with the superposition operation upstream and downstream until the right solution is found and measured with a probability equal to 1.

7. Proposed algorithm

In this section, we present the generalized principle of the quantum approach adapted to the solution of combinatorial problems.

In the following we consider that the given combinatorial problem has at least one good solution, namely $|w\rangle$, among N possible solutions $|0\rangle, |1\rangle, \dots, |w\rangle, \dots, |N-1\rangle$.

The Grover algorithm (see Section 6) for finding the correct solution starts by making a uniform superposition, as shown in Fig. 5. The search space is materialized by $\{|0\rangle, |1\rangle, \dots, |w\rangle, \dots, |N-1\rangle\}$ or simply $\{0, 1, \dots, w, \cdot, N-1\}$. We can notice the uniform superposition by the same amplitude level $1/\sqrt{N}$ of all solutions in the figure.

Thus, unlike the algorithm of Grover, which consists of making first a uniform superposition of all potential solutions, namely $|0\rangle, |1\rangle, \dots,$ $|w\rangle, \dots, |N-1\rangle$, as in Fig. 5, our algorithm makes a non-uniform superposition, depending on the area where we think we have the right solution.

Compared to Figs. 5, 6, makes a non-uniform superposition of potential solutions. In this example, we note that the correct solution is in the second half of the search space on the right. So, instead of doing a uniform superposition, we proposed a non-uniform superposition by emphasizing on the second search area and initially increasing the amplitudes of the solutions. We can observe in Fig. 6 that the amplitudes of the solutions in the good search area are the double of the solutions in the rest of the search space. In addition, the amplitudes in the rest of the search space can even be canceled to reduce the search space to the selected area (see Fig. 7).

The second step of the algorithm consists in increasing the amplitude of the sought solution and decreasing the amplitude of the other solutions. But before increasing the amplitude of the good solution, in this case $|w\rangle$, the quantum oracle U_w is applied to the quantum state of the initial superposition. The quantum oracle allows one to determine

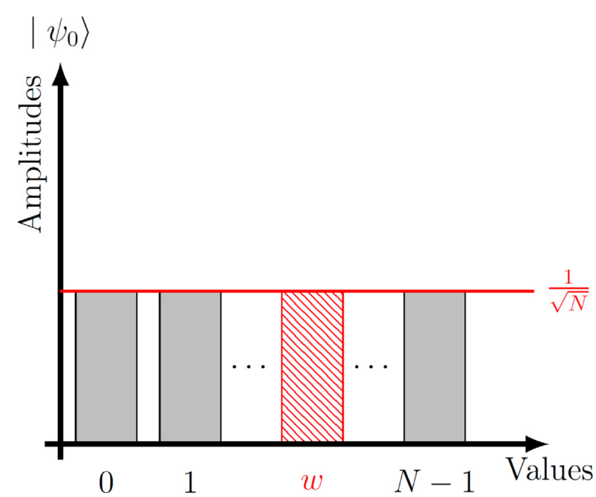

Fig. 5. Uniform superposition of the initial solutions.

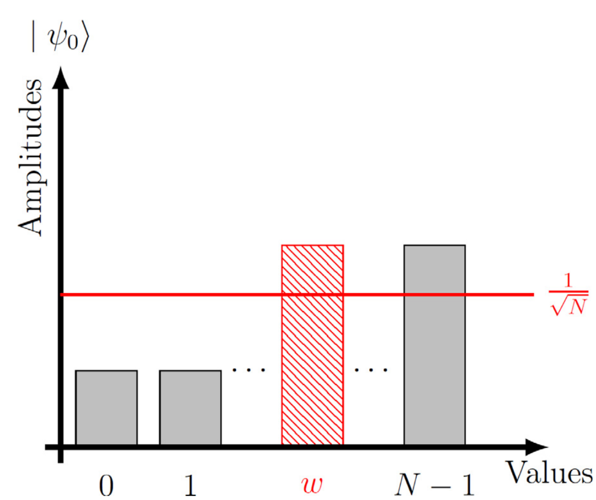

Fig. 6. Nonuniform superposition of our approach.

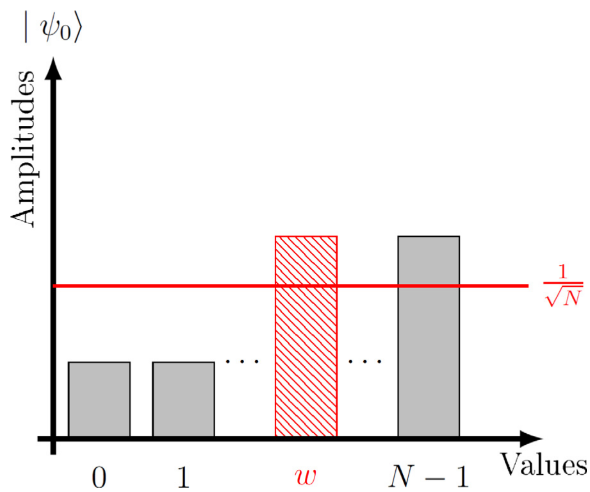

Fig. 7. Non-uniform superposition of our approach with cancellation.

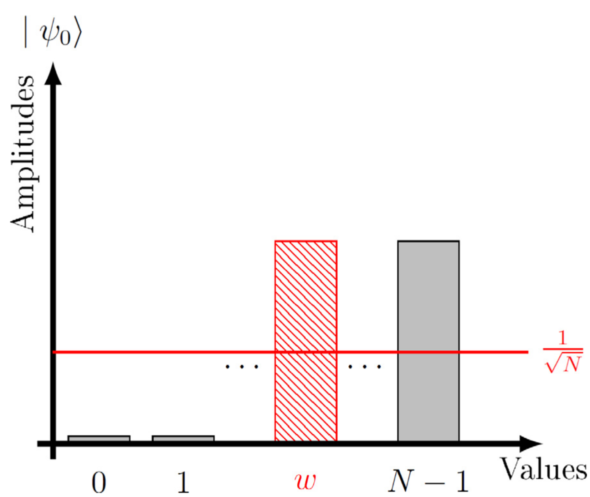

Fig. 8. The right solution marked by the oracle.

whether a given solution is the sought one or not. If a solution is the sought one, the oracle marks it by making a phase inversion (see Fig. 8).

The third step is to increase the amplitude of the desired solution. This quantum process, which is carried out by means of the Grover operator, consists of an inversion or rotation of the sought solution with respect to the average of the amplitudes, as described in Fig. 9.

This last rotation consists of bringing the current state closer to the sought solution $|w\rangle$. The amplitude corresponding to $|w\rangle$ increases compared to its initial value. The last two rotations performed, to mark the desired solution and to increase its amplitude, constitute a single iteration and are repeated several times up to $\mathcal{O}(\sqrt{N/S})$ to find a good solution with a probability of 1. N is the number of elements or initial solutions and S the number of subdivisions or the reduction factor.

Summary

1. In the proposed approach, based on Grover's algorithm, a uniform superposition is replaced by a non-uniform superposition.
2. Due to the observations made in Tables 3 and 4, in Section 5, the uniform superposition is mitigated by canceling the amplitude of some solutions and increasing the amplitude of other ones. This acts as a reduction or subdivision of the initial search space.

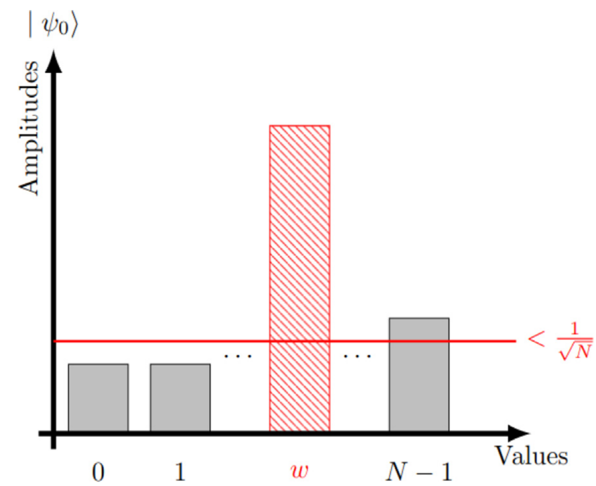

Fig. 9. Rotation around the average of the amplitudes.

3. Our reduction or subdivision of the search space always respects the unitarity of the quantum state.

7.1. Generating amplitudes

In order to have a robust Grover's algorithm, several authors have used a non-uniform superposition of the amplitudes of basis states. Biham et al. (1999) proposed and analyzed an input with non-uniform random amplitudes. Their approach allows a non-uniform but random superposition, which is not suitable to define the input of our algorithm. Ventura and Martinez (1999) proposed a superposition of some known values of a function. Their approach is more adapted to machine learning algorithms. It allows the superposition of only known values of a function to determine or learn from the function that is unknown. In this approach, initialization is done using two inputs: the register that contains the known examples or values to be superposed and the values of the function corresponding to these known examples. The function values are used as coefficients for the corresponding states. However, in our case, we do not know beforehand the function values corresponding to the inputs. Therefore, a non-uniform amplitude generator is required that allows one to automatically reduce the search space by canceling the amplitude of some known values. The idea is to search only for one of the areas: $-\infty, +\infty$, or the mean, while satisfying the unitarity condition. The following equation is used to increase the amplitude of elements in the restricted search.

$$|\psi_{area}\rangle = \frac{1}{\sqrt{n_S}} \sum_{x=i}^{N_a} |x\rangle \tag{19}$$

where $i = K \times n_S$, with $K = 0, 1, \dots S-1, n_S = \frac{N}{S}, N_a = i + n_S$ and $N = 2^n$. Noting that n is the number of qubits, n_S is the number of elements in the new reduced search area, S is the number of reduced quantum areas or substates.

7.2. Pseudocode of the proposed algorithm

The pseudocode of the proposed algorithm is given by Algorithm 1. It consists mainly of two parts. The first part (the classical part) starts by counting the number of positive (pos) and negative (neg) variables in all the clauses and then computes the ratio $r = pos/neg$. The search area is then identified according to the value of r. The second part (the quantum part) increases the amplitudes of the solutions in the selected area and applies Grover's algorithm.

Algorithm 1 Quantum Hybrid algorithm (K-SAT, K, num-clauses).

```
positive, negative : integer;
positive = 0;
negative = 0;
for clause in K-SAT do
    for variable in clause do
        if variable > 0 then
            positive = positive + 1;
        else
            negative = negative + 1;
        end if
    end for
end for
Compute the ratio r = positive/negative;
Subdivide the search space;
if r > 2 then
    AmplifierDivisor(numOfQubits, +∞);
    Apply the Grover search on the space +∞;
else
    if r < 0.5 then
        AmplifierDivisor(numOfQubits, −∞);
        Apply the Grover search on the space −∞;
    else
        AmplifierDivisor(numOfQubits, mean);
        Apply the Grover search on the space mean
    end if
end if
```

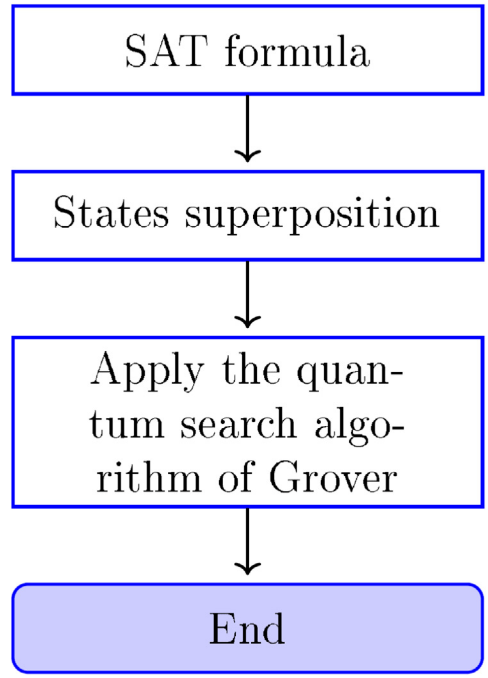

Fig. 10. Original Grover algorithm flow diagram (Grover, 1996).

The original flow diagram proposed by Grover is shown in Fig. 10, while Fig. 11 shows the flow diagram of the proposed one.

7.3. Complexity of our approach

The complexity of the proposed algorithm depends on the modified Grover algorithm with non-uniform input, which is the quantum part. The proposed algorithm consists mainly of two parts, a classical part and a quantum part, which uses the modified Grover quantum search with a non-uniform initial superposition. The classical part, which consists of counting the number of positive and negative variables, can be done polynomially, using kC operations. With k the number of variables in each clause and C the number of clauses. Therefore, with N being the number of solutions and S the number of subdivisions or the reduction factor of the search space, the complexity is $\mathcal{O}(\sqrt{N/S})$. If we have l good solutions among the N possible solutions, the complexity becomes $\mathcal{O}(\sqrt{N/lS})$

8. Results and discussions

To test the performance of the proposed algorithm, it is implemented concurrently with the original Grover algorithm. The tests are performed on the 32 qubits IMB QASM quantum simulator and the results are obtained after 1024 shots on an Intel© Pentium© CPU 2020M @ 2.40 GHz × 2, memory 4 GiB, with a Linux Mint 20.1 Cinnamon as operating system.

Tests are performed on SAT problems with 4 variables. The limitation in terms of variables is due to the QASM simulator that can support up to only 5 qubits. With these 4 variables, the two tested algorithms need a quantum register of 4 qubits to encode all solutions of the SAT problem. To the best of our knowledge, the test performed is the first test on SAT problems that exceeded 3 variables (Wang, 2021).

The tests are performed on two different SAT problems of different categories. The first problem is a randomly generated SAT problem and the second one is a Horn-SAT problem. The first SAT problem considered for the test is the SAT formula of Eq. (20) and the Horn-SAT problem is in Eq. (21).

$$F1(x_1,x_2,x_3,x_4) = (x_1 \vee \neg x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3 \vee \neg x_4) \wedge \\ (\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3 \vee x_4) \tag{20}$$

$$F_2(x_1,x_2,x_3,x_4) = (x_1 \vee \neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4) \wedge \\ (\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \tag{21}$$

The SAT formula of the problem is used as input to our algorithm. The first step counts the number of positive and negative variables and computes the ratio r. Since most variables are positive and negative, respectively, in F_1 and F_2, the ratio r is higher than 2 (with $r = 13/3 = 4.333333333$) in F_1 and lower than $\frac{1}{2}$ (with $r = 4/12 = 0.25$) in F_2. The selected spaces are the $+\infty$ space and the $-\infty$ space for F_1 and F_2, respectively.

The input of the proposed algorithm is a non-uniform superposition of basic solutions. Fig. 12 shows that after applying our splitting amplifier on the 4 qubits quantum state, we obtain a non-uniform superposition of all the initial solutions. As the majority of our variables are positive, it increases the amplitudes of the solutions in the last third part of the search area and sets to zero the amplitudes of the solutions in the rest of the search area (see Fig. 13).

The original input of the Grover algorithm is a uniform superposition of all possible solutions (see Fig. 14). The original Grover algorithm does not care about the position of the solution, but considers all the solutions equally in the search space.

Running the algorithm with each input for a single iteration, we get the results shown in Figs. 15 and 16, and 17 and 18 for F_1 and F_2, respectively.

In these results represented by the graphs in Figs. 15, 16, 17 and 18 the binary numbers on four bits $0000, 0001, \ldots, 1111$ in the abscissa axis represent the possible basic solutions and the ordinate axis represents the probabilities to find each solution when one carries out a measurement on the quantum state constituted by the superposition of all the possible solutions. Since quantum algorithms manipulate quantum

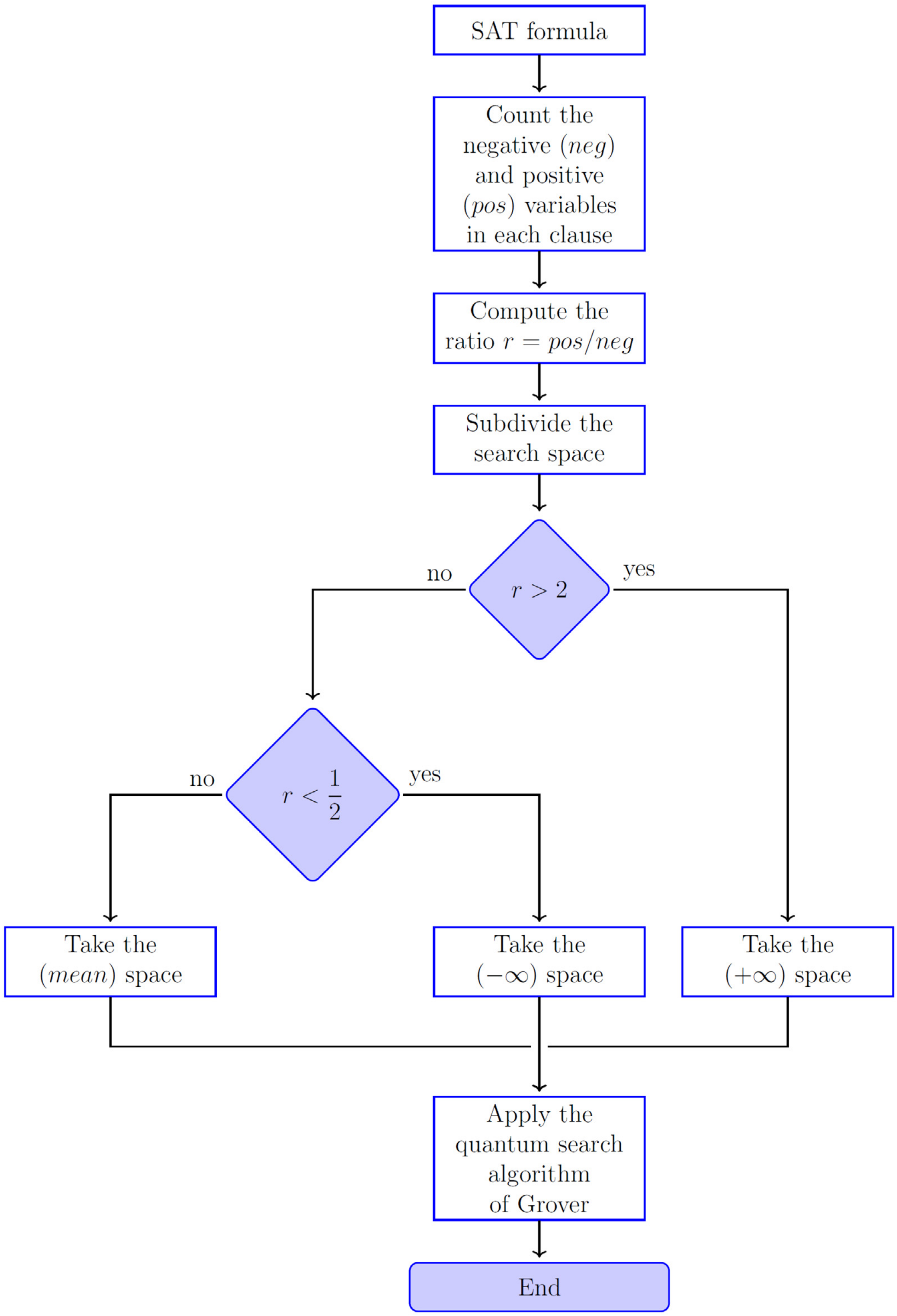

Fig. 11. Proposed algorithm flowchart: taking as input a SAT formula, it starts by counting the number of positive and negative variables. Then it computes the ratio r. Based on r it decides the right area to search for a solution and applies the Grover's quantum search.

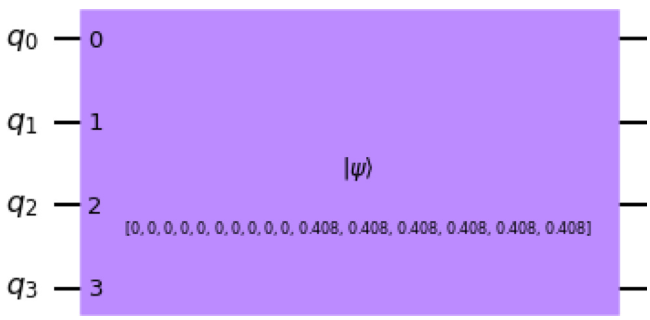

Fig. 12. Improved input of the proposed algorithm for the F_1 formula.

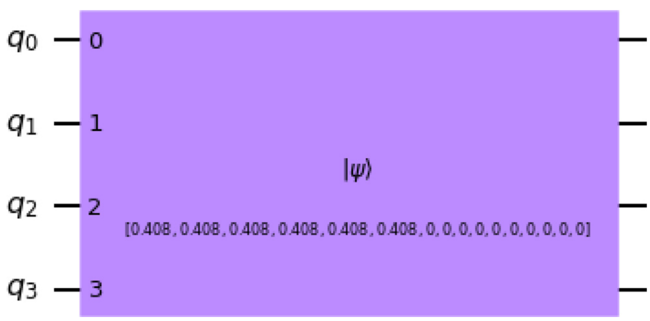

Fig. 13. Improved input of the proposed algorithm for the F_2 formula.

states, the objective of quantum algorithms will be to avoid performing measurements throughout the computational process except at the end of the calculations.

The good solution is the binary number with the highest probability. Thus, it appears from the graphs that a good solution for F_1 is 1010 because in the two graphs, Figs. 15 and 16 this solution is the one with the highest probability. Respectively, 0.114 for the SAT algorithm based on the original input, and 0.272 for the SAT algorithm based on the improved input we proposed.

The same way for the Horn-SAT, F_2, the good solution is 0000 because on the two graphs, Figs. 17 and 18 representing the two algorithm tests, this solution is the one with the highest probability.

We observe that the probability of finding the correct solution with our approach with a single iteration is twice the probability of finding the correct solution of SAT using the Grover algorithm with uniform superposition input. This also means that if the time complexity of the Grover algorithm for SAT is $\mathcal{O}(\sqrt{N})$ with the original input, then the time complexity of the Grover algorithm will be $\mathcal{O}(\sqrt{N/2})$ with the improved input in our case. This is because we have subdivided the search space into two, taking the second half of the search space, due to the dominance of positive and negative variables in F_1 and the Horn-SAT respectively.

It also appears that the probability of the correct solution, with one iteration only, depends on the number of solutions in the space, and is higher if the solution is unique. This is how we can observe the difference in probability between the two tests on F_1 and F_2. For the Horn-SAT case, F_2, all the solutions in the selected subspace are good. This is why the probability of a good solution is low compared to the results on the random SAT, F_1.

In terms of computation time, the differences between the two tests are of the order of a few seconds. On the two tests performed, Grover's algorithm takes 1 m 33 s, and the proposed algorithm takes 1m2s on average. This is justified. The proposed algorithm first targets the search area, but as soon as the targeting is completed, the number of solutions in the targeted area is very few, so easy to search the right one. For example, in the tests performed, in the targeted area, in both, contained only 6 solutions among which the right one, while with the uniform distribution we have a total of 16 solutions among which to search the right one.

Compared to the results of Cheng and Tao (2007) our algorithm is faster since the complexity is $\mathcal{O}(\sqrt{N/lS})$ versus $\mathcal{O}(\sqrt{N/l})$ as previously proved. However, the proposed algorithm requires more qubits. The proposed algorithm uses the same number of qubits as the results (Yang et al., 2009; Wang et al., 2020; Mandra et al., 2016) but the number of possible solutions in the search space is smaller.

Case with no solution

Fig. 19 shows the behavior of amplitudes in the case where there is no solution to the SAT problem in the search area. Considering an SAT problem where the right solution is in the first index, if we focus the search in the second index, we observe that amplitudes remain practically at the same level. This gives us the possibility in the framework to know when a combinatorial problem does not admit a solution. This is due to the fact that the oracle does not mark any potential solution as good, that is, $f(x) = 0$ for each variable.

9. Conclusion

This work introduced an efficient algorithm for solving combinatorial problems based on quantum properties such as superposition of

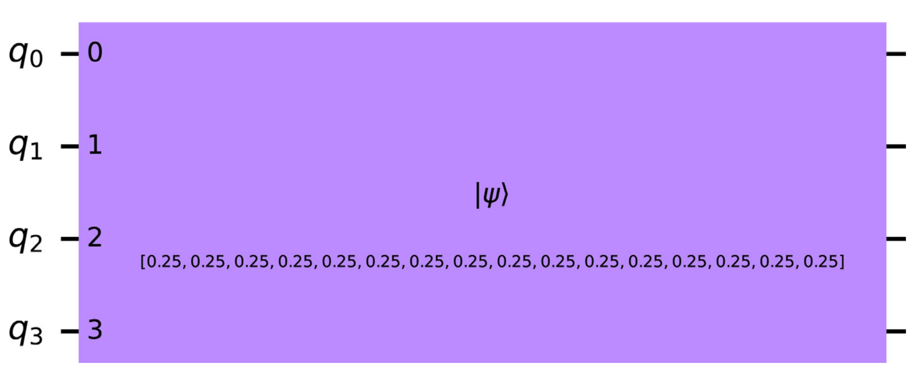

Fig. 14. Uniform input of the Grover algorithm.

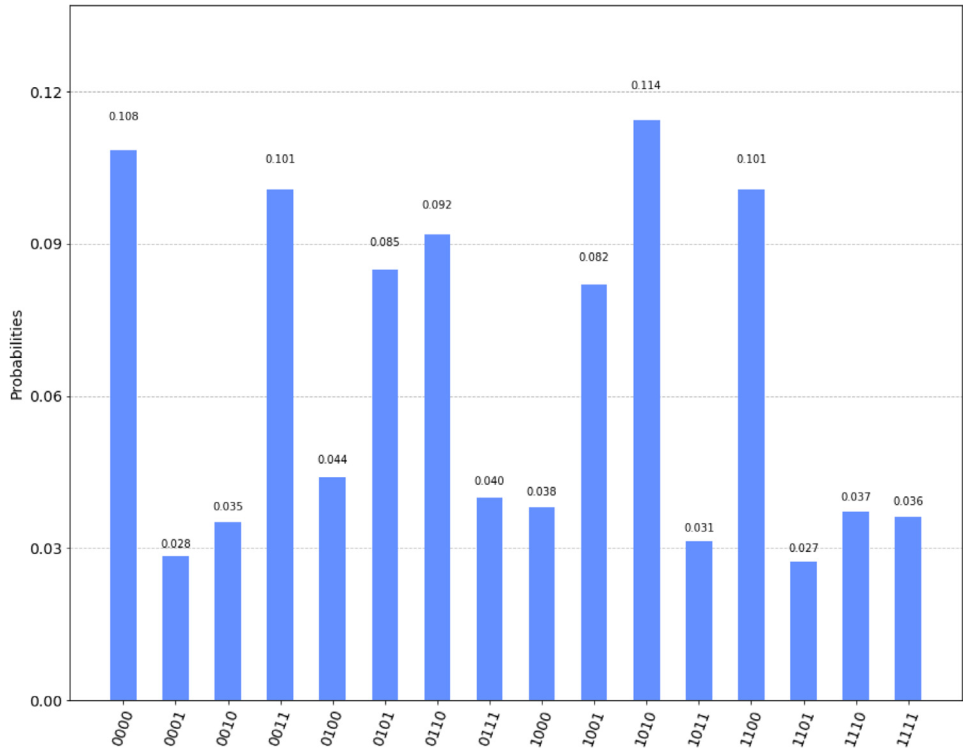

Fig. 15. Result of Grover algorithm with uniform input for one iteration on F_1.

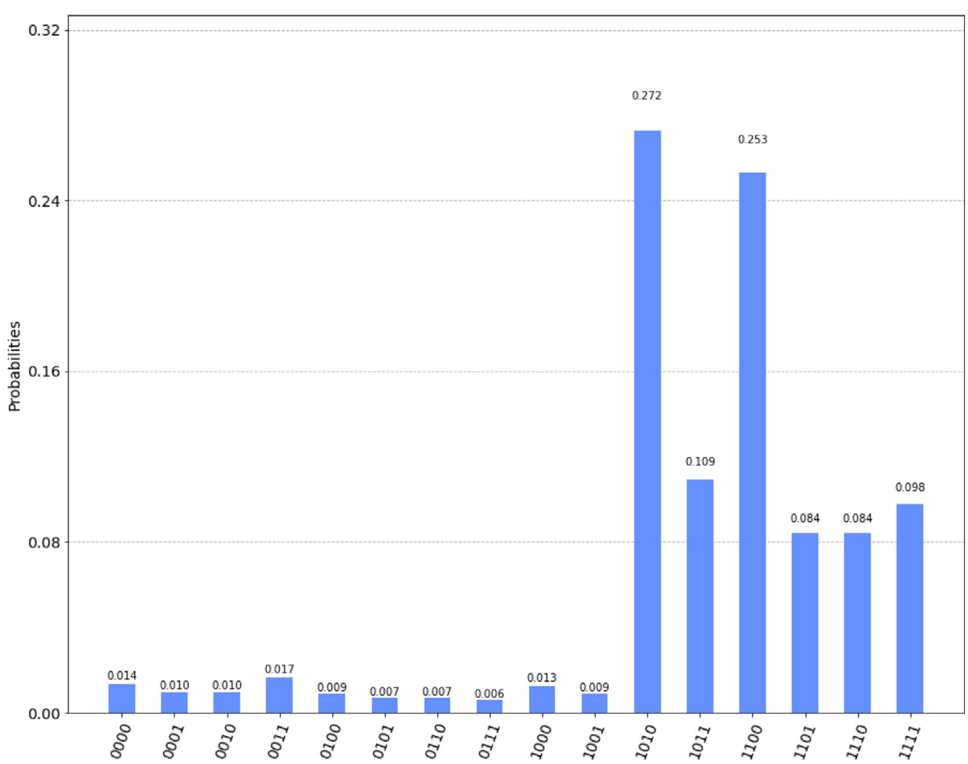

Fig. 16. Result of the proposed algorithm with improved input for one iteration in F_1.

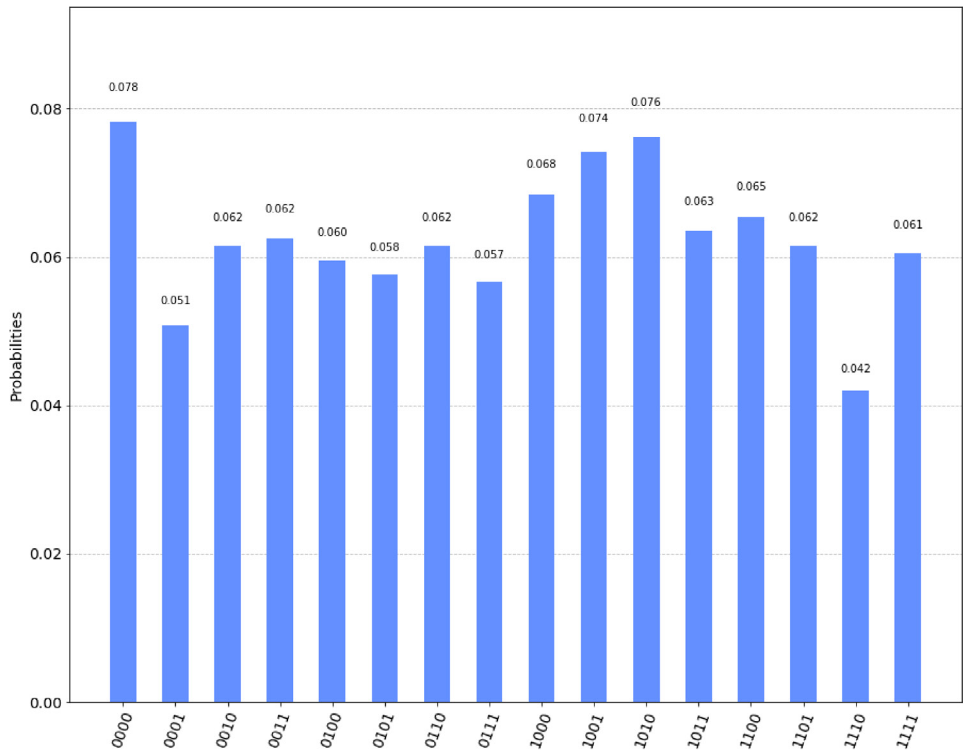

Fig. 17. Result of the Grover algorithm with uniform input for one iteration on Horn-SAT.

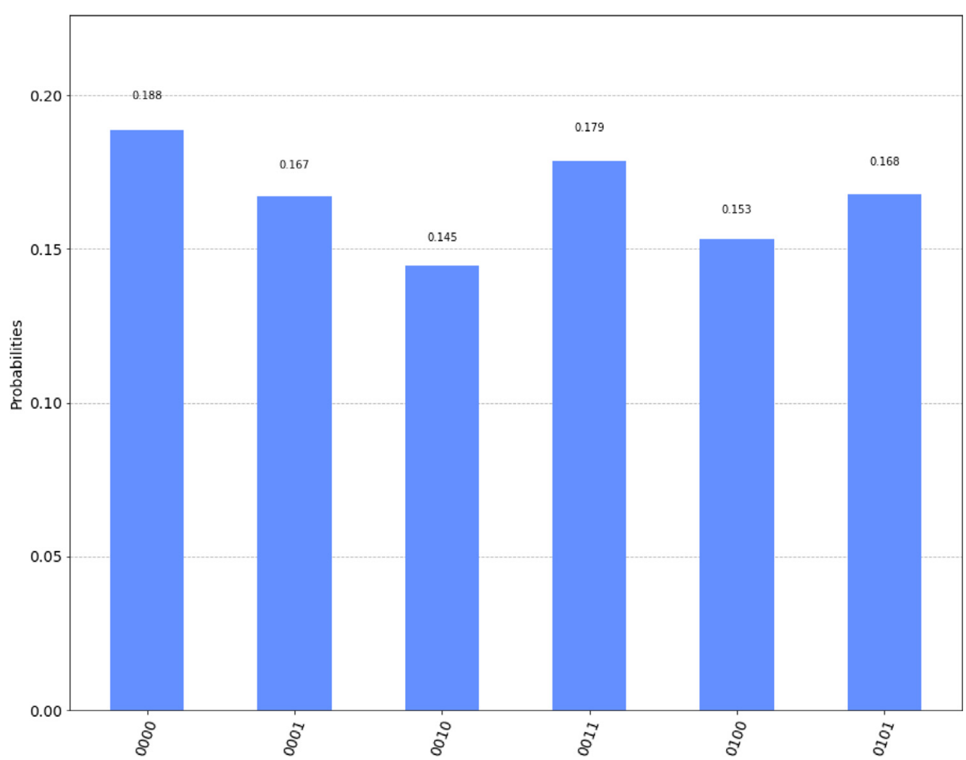

Fig. 18. Result of the proposed algorithm with improved input for one iteration on Horn-SAT.

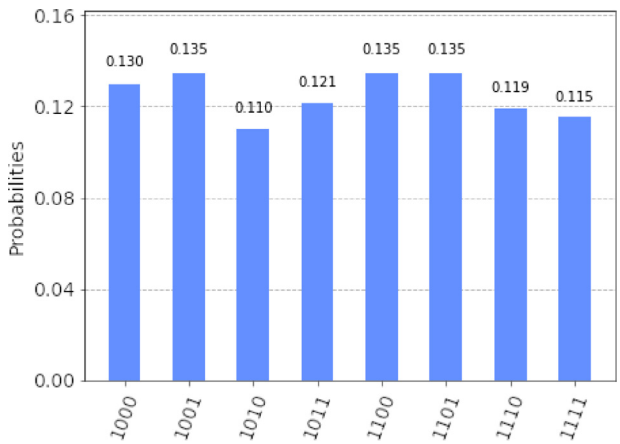

Fig. 19. Result of the Grover algorithm with our improved input when there is no solution.

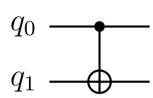

Fig. A.20. `CNOT` gate.

states. Combinatorial problems are problems that have a very large search space for the solution. The proposed algorithm uses a technique of reduction or subdivision of the search space in order to reduce the number of computations. This is done by introducing a uniform amplitude perturbation that aims to reduce or cancel the amplitudes of some solutions while increasing the amplitudes of the others in the selected area. The reduction and subdivision of the space allow us to introduce data parallelism in the computation. In fact, the chance of finding a good solution in a shorter time can be increased by targeting several search areas and applying several local searches in the different areas.

The proposed algorithm has been applied to the satisfaction problem (SAT). The results showed that the proposed algorithm is able to find the right solution with a reduction in terms of complexity, that is, $\mathcal{O}(\sqrt{N/S})$ against $\mathcal{O}(\sqrt{N})$, with N being the number of elements and S the subdivision or reduction factor of the search space. Moreover, if we have l good solutions among the N possible solutions, our complexity becomes $\mathcal{O}(\sqrt{N/lS})$.

The smaller the selected area, the faster the algorithm. But a trade-off must be found to avoid the algorithm returning no solution to a problem that has one, just because the selected area is not the right one. Therefore, other approaches must be developed to determine an optimal reduction and selection of the promising area. Moreover, in large problems, several promising areas could exist. It would therefore be necessary to determine these areas to ensure that the algorithm will find a solution in case it exists. Furthermore, large search space test sizes can be performed to ensure scalability and robustness, using the same approach as in Cheng and Tao (2007) can be conducted.

One of the limitations of the proposed algorithm is that the ratio used seems not to be adapted for the XORSAT problem. In fact, the zone targeting used here by means of the ratio does not allow us to delimit the different suitable zones for the XORSAT as we have studied above for other SAT instances. Another interesting perspective will be the adaptation of this technique to the XORSAT problem.

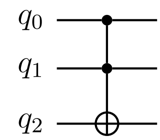

Fig. A.21. Toffoli gate.

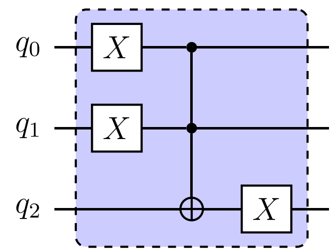

Fig. A.22. Circuits that compute $(a \wedge b)$ and $(a \vee b)$.

CRediT authorship contribution statement

Charles Moudina Varmantchaonala: Conceptualization, Methodology, Software, Writing – original draft. **Jean Louis Kedieng Ebongue Fendji:** Conceptualization, Methodology, Validation, Writing – review & editing, Supervision. **Jean Pierre Tchapet Njafa:** Methodology, Validation, Writing – review & editing. **Marcellin Atemkeng:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

We thank Pr. Serge Guy NANA ENGO for his helpful discussions and remarks.

Appendix A. Encoding SAT formula

Qiskit is an open-source framework developed by IBM[1,2] to simulate quantum. It helps in implementing circuits, pulses, and algorithms. With Qiskit, it is possible to write a code that models the circuit of the algorithm that finds the correct answer to the SAT problem. A SAT clause can be expressed using basic logic gates of Figs. A.20–A.22. In this way, all the different parts of the circuit that are used to solve the SAT problem can be implemented.

[1] https://quantum-computing.ibm.com/. https://qiskit.org/documentation/index.html.
[2] https://qiskit.org/documentation/index.html.

Appendix B. AmplifierDivisor module for generating amplitudes

The following code presents our used AmplifierDivisor() module, which generates the amplitudes.

```
def Amplificateur_Diviseur(num_of_qubits,num_sub_states):

    #n=QuantumRegister(num_of_qubits)
    subsets = np.empty(num_sub_states, dtype=object)
    N=2**num_of_qubits
    index = 0
    sup_index = (N//num_sub_states)


    if (N%num_sub_states != 0):
        k=0
        for i in range(1,num_sub_states):
            sup = [0.]*N # all norm initialize to zero
            num_el = (N//num_sub_states) + 1
            sub_state=QuantumCircuit(num_of_qubits)

            for j in range(index, sup_index+1):
                #creation of the list where the 0s are at indices on the
                #first part you don't want
                sup[j]=np.sqrt((N/num_el)/N)

            sub_state.initialize(sup, range(num_of_qubits))
            subsets[k]= sub_state
            #progression au suivant subset
            index = index + (N//num_sub_states) + 1
            sup_index = sup_index + (N//num_sub_states) +1
            k=k+1

        #last one subset
        sub_state=QuantumCircuit(num_of_qubits)
        sup = [0.]*N # all norm initialize to zero
        for j in range(index-1,sup_index):
            #creation of the list where the 0s are at indices on the
            #first part you don't want
            sup[j]=np.sqrt((N/num_el)/N)
            #print(sup[j])

        sub_state.initialize(sup, range(num_of_qubits))
        subsets[num_sub_states-1]= sub_state

    else:
        k=0
        for i in range(0,num_sub_states):

            #initialisation du ie subset for j in range(index-1,sup-1)
            sup = [0.]*N # all norm initialize to zero
            num_el = N/num_sub_states
            sub_state=QuantumCircuit(num_of_qubits)

            for j in range(index,sup_index):
                #creation of the list where the 0s are at indices on
                #the first part you don't want
                sup[j]=np.sqrt((N/num_el)/N)

            sub_state.initialize(sup, range(num_of_qubits))
            subsets[k] = sub_state

            #progression to the next subset
            index = index + (N//num_sub_states)
            sup_index = sup_index + (N//num_sub_states)
            k=k+1
    return subsetsBMS
```

References

Achlioptas, D., Beame, P., Molloy, M., 2004. Exponential bounds for DPLL below the satisfiability threshold. In: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Citeseer, pp. 139–140.

Apolloni, B., Carvalho, C., De Falco, D., 1989. Quantum stochastic optimization. Stochastic Process. Appl. 33 (2), 233–244, Publisher: Elsevier.

Battaglia, D.A., Santoro, G.E., Tosatti, E., 2005. Optimization by quantum annealing: Lessons from hard satisfiability problems. Phys. Rev. E 71 (6), 066707, Publisher: APS.

Biham, E., Biham, O., Biron, D., Grassl, M., Lidar, D.A., 1999. Grover's quantum search algorithm for an arbitrary initial amplitude distribution. Phys. Rev. A 60 (4), 2742, Publisher: APS.

Bresler, G., Huang, B., 2022. The algorithmic phase transition of random k-sat for low degree polynomials. In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS). IEEE, pp. 298–309.

Bryant, R.E., Heule, M.J., 2021. Generating extended resolution proofs with a BDD-based SAT solver. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, pp. 76–93.

Campos, E., Venegas-Andraca, S.E., Lanzagorta, M., 2021. Quantum tunneling and quantum walks as algorithmic resources to solve hard K-SAT instances. Sci. Rep. 11 (1), 1–18, Publisher: Nature Publishing Group.

Cheng, S.-T., Tao, M.-H., 2007. Quantum cooperative search algorithm for 3-SAT. J. Comput. System Sci. 73 (1), 123–136, Publisher: Elsevier.

Chou, C.-N., Love, P.J., Sandhu, J.S., Shi, J., 2022. Limitations of local quantum algorithms on random max-k-xor and beyond. In: 49th International Colloquium on Automata, Languages, and Programming (ICALP 2022). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Chvátal, V., Reed, B., 1992. Mick gets some (the odds are on his side)(satisfiability). In: Proceedings., 33rd Annual Symposium on Foundations of Computer Science. IEEE Computer Society, pp. 620–627.

Coja-Oghlan, A., Haqshenas, A., Hetterich, S., 2017. Walksat stalls well below satisfiability. SIAM J. Discrete Math. 31 (2), 1160–1173, Publisher: SIAM.

Cook, S.A., 1971. The complexity of theorem-proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158.

de Falco, D., Tamascelli, D., 2011. An introduction to quantum annealing. RAIRO-Theor. Inform. Appl. 45 (1), 99–116, Publisher: EDP Sciences.

Dixit, A.A., Kolaitis, P.G., 2019. A SAT-based system for consistent query answering. In: International Conference on Theory and Applications of Satisfiability Testing. Springer, pp. 117–135.

Egger, D.J., Mareek, J., Woerner, S., 2021. Warm-starting quantum optimization. Quantum 5, 479, Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

Erós, E., Dahl, M., Falkman, P., Bengtsson, K., 2021. Evaluation of high level methods for efficient planning as satisfiability. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, pp. 1–8.

Farhi, E., Goldstone, J., Gutmann, S., 2002. Quantum adiabatic evolution algorithms versus simulated annealing. arXiv preprint Quant-Ph/0201031.

Farhi, E., Goldstone, J., Gutmann, S., 2014. A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.

Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., Preda, D., 2001. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science 292 (5516), 472–475, Publisher: American Association for the Advancement of Science.

Farhi, E., Goldstone, J., Gutmann, S., Neven, H., 2017. Quantum Algorithms for Fixed Qubit Architectures. http://dx.doi.org/10.48550/arXiv.1703.06199, URL http://arxiv.org/abs/1703.06199 arXiv:1703.06199 [quant-ph].

Farhi, E., Harrow, A.W., 2016. Quantum supremacy through the quantum approximate optimization algorithm. arXiv preprint arXiv:1602.07674.

Fernandes, D., Dutra, I., 2019. Using Grover's search quantum algorithm to solve Boolean satisfiability problems: Part I. XRDS: Crossroads, ACM Mag. Stud. 26 (1), 64–66, Publisher: ACM New York, NY, USA.

Franco, J., Paull, M., 1983. Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem. Discrete Appl. Math. 5 (1), 77–87, Publisher: Elsevier.

Froleyks, N., Heule, M., Iser, M., Järvisalo, M., Suda, M., 2021. SAT competition 2020. Artificial Intelligence 301, 103572, Publisher: Elsevier.

Gaber, L., Hussein, A.I., Mahmoud, H., Mabrook, M.M., Moness, M., 2020. Computation of minimal unsatisfiable subformulas for SAT-based digital circuit error diagnosis. J. Ambient Intell. Humaniz. Comput. 1–19, Publisher: Springer.

Gamarnik, D., Sudan, M., 2017. Performance of sequential local algorithms for the random NAE-K-SAT problem. SIAM J. Comput. 46 (2), 590–619, Publisher: SIAM.

Gasquet, O., Longin, D., Maris, F., Régnier, P., Valais, M., 2018. Compact tree encodings for planning as QBF. Inteligencia Artif. (Ibero-Am. J. Artif. Intell.) 21 (62), 103–113, Publisher: Asociación Española de Inteligencia Artificial (AEPIA).

Goto, H., Tatsumura, K., Dixon, A.R., 2019. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems. Sci. Adv. 5 (4), eaav2372, Publisher: American Association for the Advancement of Science.

Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212–219.

Guerreschi, G.G., Matsuura, A.Y., 2019. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. Sci. Rep. 9 (1), 1–7, Publisher: Nature Publishing Group.

Hansen, T.D., Kaplan, H., Zamir, O., Zwick, U., 2019. Faster k-sat algorithms using biased-ppsz. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. pp. 578–589.

He, F., Gao, Y., Yin, L., 2018. Efficient software product-line model checking using induction and a SAT solver. Front. Comput. Sci. 12 (2), 264–279, Publisher: Springer.

Hetterich, S., 2016. Analysing survey propagation guided decimation on random formulas. arXiv preprint arXiv:1602.08519.

Ishtaiwi, A., Alshahwan, F., Jamal, N., Hadi, W., AbuArqoub, M., 2021. A dynamic clause specific initial weight assignment for solving satisfiability problems using local search. Algorithms 14 (1), 12, Publisher: MDPI.

Jabbour, S., Mhadhbi, N., Raddaoui, B., Sais, L., 2020. SAT-based models for overlapping community detection in networks. Computing 102 (5), 1275–1299, Publisher: Springer.

Kasihmuddin, M.S.M., Mansor, M.A., Sathasivam, S., 2018. Satisfiability based reverse analysis method in diabetes detection. In: AIP Conference Proceedings, Vol. 1974. AIP Publishing LLC, 020020, Issue: 1.

Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y., Balaprakash, P., 2020. Learning to optimize variational quantum circuits to solve combinatorial problems. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. pp. 2367–2375, Issue: 03.

Khurshid, S., Marinov, D., 2004. TestEra: Specification-based testing of Java programs using SAT. Autom. Softw. Eng. 11 (4), 403–434, Publisher: Springer.

King, A.D., McGeoch, C.C., 2014. Algorithm engineering for a quantum annealing platform. arXiv preprint arXiv:1410.2628.

Kinjo, M., Sato, S., Nakamiya, Y., Nakajima, K., 2005. Neuromorphic quantum computation with energy dissipation. Phys. Rev. A 72 (5), 052328, Publisher: APS.

Konovalenko, A., 2019. Developing a Heuristic Algorithm for Classification of Problems with Binary Attributes (Master's thesis). Høgskolen i Molde-Vitenskapelig høgskole i logistikk.

Leporati, A., Felloni, S., 2007. Three "quantum" algorithms to solve 3-SAT. Theoret. Comput. Sci. 372 (2–3), 218–241, Publisher: Elsevier.

Lin, J., Lai, Z.Y., Li, X., 2020. Quantum adiabatic algorithm design using reinforcement learning. Phys. Rev. A 101 (5), 052327, Publisher: APS.

Luo, N., Judson, S., Antonopoulos, T., Piskac, R., Wang, X., 2022. ${$ppSAT$}$: Towards ${$Two-Party$}$ Private ${$SAT$}$ Solving. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 2983–3000.

Mandra, S., Guerreschi, G.G., Aspuru-Guzik, A., 2016. Faster than classical quantum algorithm for dense formulas of exact satisfiability and occupation problems. New J. Phys. 18 (7), 073003, Publisher: IOP Publishing.

McGeer, P.C., Saldanha, A., Stephan, P.R., Brayton, R.K., Sangiovanni-Vincentelli, A.L., 1991. Timing analysis and delay-fault test generation using path-recursive functions. In: 1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers. IEEE Computer Society, pp. 180–181.

Ming-Te, C., Franco, J., 1990. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k satisfiability problem. Inform. Sci. 51 (3), 289–314, Publisher: Elsevier.

Munoz, J.G., Delgado, F., 2016. QUANTUM: a wolfram mathematica add-on for Dirac bra-ket notation, non-commutative algebra, and simulation of quantum computing circuits. In: Journal of Physics: Conference Series, Vol. 698. IOP Publishing, 012019, Issue: 1.

Nam, W., Lee, D., Kil, H., 2018. Performance Analysis of Satisfiability Solvers for Bounded Model Checking. Int. Inf. Inst. (Tokyo). Inf. 21 (5), 1583–1594, Publisher: International Information Institute.

Paturi, R., Pudlák, P., Zane, F., 1997. Satisfiability coding lemma. In: Proceedings 38th Annual Symposium on Foundations of Computer Science. IEEE, pp. 566–574.

Sathasivam, S., Mansor, M.A., Kasihmuddin, M.S.M., Abubakar, H., 2020. Election algorithm for random k satisfiability in the Hopfield neural network. Processes 8 (5), 568, Publisher: MDPI.

Somma, R.D., Boixo, S., Barnum, H., Knill, E., 2008. Quantum simulations of classical annealing processes. Phys. Rev. Lett. 101 (13), 130504, Publisher: APS.

Stava, M., 2021. A synthesis of reversible digital circuits to solve the Boolean satisfiability. In: Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems. Springer, pp. 233–246.

Steffen, M., van Dam, W., Hogg, T., Breyta, G., Chuang, I., 2003. Experimental implementation of an adiabatic quantum optimization algorithm. Phys. Rev. Lett. 90 (6), 067903, Publisher: APS.

Tucker, C., Shuffelton, D., Jhala, R., Lerner, S., 2007. Opium: Optimal package install/uninstall manager. In: 29th International Conference on Software Engineering (ICSE'07). IEEE, pp. 178–188.

Ventura, D., Martinez, T., 1999. Initializing the amplitude distribution of a quantum state. Found. Phys. Lett. 12 (6), 547–559, Publisher: Springer.

Wang, R., 2021. Comparing Grover's quantum search algorithm with classical algorithm on solving satisfiability problem. In: 2021 IEEE Integrated STEM Education Conference (ISEC). IEEE, p. 204.

Wang, P., Liu, G., Liu, L., 2020. A generic variable inputs quantum algorithm for 3-sat problem. In: 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). IEEE, pp. 308–312.

Yang, W.-L., Wei, H., Zhou, F., Chang, W.-L., Feng, M., 2009. Solution to the satisfiability problem using a complete Grover search with trapped ions. J. Phys. B: At. Mol. Opt. Phys. 42 (14), 145503, Publisher: IOP Publishing.

Zhang, R., Chen, J., Zhao, H., 2020. Procedure of Solving 3-SAT Problem by Combining Quantum Search Algorithm and DPLL Algorithm. Computing 4, 14–24.

Zhu, L., Tang, H.L., Barron, G.S., Calderon-Vargas, F.A., Mayhall, N.J., Barnes, E., Economou, S.E., 2022. Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. Phys. Rev. Res. 4 (3), 033029, Publisher: APS.