Solving systems of Boolean multivariate equations with quantum annealing

Sergi Ramos-Calderer,[1,2] Carlos Bravo-Prieto,[1,2] Ruge Lin,[1,2] Emanuele Bellini,[3] Marc Manzano,[4,5] Najwa Aaraj,[3] and José I. Latorre[1,2,6]

[1]*Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, United Arab Emirates*
[2]*Departament de Física Quàntica i Astrofísica and Institut de Ciències del Cosmos (ICCUB), Universitat de Barcelona, Martí i Franquès 1, 08028 Barcelona, Spain*
[3]*Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, United Arab Emirates*
[4]*Sandbox@Alphabet, Mountain View, California, USA*
[5]*Electronics and Computing Department, Faculty of Engineering, Mondragon Unibertsitatea, Mondragon, Spain*
[6]*Centre for Quantum Technologies, National University of Singapore, Singapore*

(Received 29 November 2021; accepted 25 January 2022; published 7 February 2022)

Polynomial systems over the binary field have important applications, especially in symmetric and asymmetric cryptanalysis, multivariate-based postquantum cryptography, coding theory, and computer algebra. In this paper, we study the quantum annealing model for solving Boolean systems of multivariate equations of degree 2, usually referred to as the multivariate quadratic problem. We present different methodologies to embed the problem into a Hamiltonian that can be solved by available quantum annealing platforms. In particular, we provide three embedding options, and we highlight their differences in terms of quantum resources. Moreover, we design a machine-agnostic algorithm that adopts an iterative approach to better solve the problem Hamiltonian by repeatedly reducing the search space. Finally, we use D-Wave devices to successfully implement our methodologies on several instances of the multivariate quadratic problem.

DOI: 10.1103/PhysRevResearch.4.013096

I. INTRODUCTION

Adiabatic quantum computation is a universal quantum computation scheme [1] where a quantum system is prepared in the ground state of an easy-to-prepare Hamiltonian and evolved towards a Hamiltonian that encodes the solution of a problem in its ground state. If the evolution is performed adiabatically, the quantum system will remain in its instantaneous ground state, and the problem will be solved. Quantum annealers are special-purpose devices based upon the principles of adiabatic quantum computation that work with simpler Hamiltonians and more relaxed evolution times. However, these devices are believed to provide an edge when solving classical satisfiability problems by leveraging quantum phenomena and are easier to control and scale up for larger, real-life problems [2–12]. Interestingly, quantum annealers on the order of thousands of qubits are already commercially available from D-Wave [13].

The quantum annealing approach to quantum computing is a research topic that can be relevant to many research problems in a variety of scientific fields. Motivated by this idea, we investigate the possibility of using the D-Wave quantum annealer for a fundamental problem in computer science: solving systems of multivariate polynomial equations over the binary field. If all polynomials in the system are linear, then the system can be efficiently solved, for instance, by Gaussian elimination. The problem is easy also when the system is either underdetermined (many fewer equations than variables), overdetermined (many more equations than variables), or sparse (the number of terms is linear with respect to the number of variables). However, the problem is known to be NP-hard already for generic quadratic systems [14]. Moreover, assuming the exponential-time hypothesis [15], there exists no subexponential time (worst-case) algorithm for this problem.

Polynomial systems over the binary field can be used to perform algebraic cryptanalysis [16] potentially against any cipher. Moreover, in the case of degree 2 polynomials, the problem, usually referred to as the multivariate quadratic (MQ) problem, has important applications in postquantum cryptography, since several postquantum schemes exist basing their security on its difficulty to be solved [17,18]. For these reasons, there is a spreading interest in the scientific community to find new algorithms to solve the MQ problem, both in the classical and quantum computation model. The former case has been extensively studied (see, for example, Refs. [19–22] for comprehensive surveys of the most effective algorithms). Regarding the latter, a detailed analysis of the required qubits and time for a Grover's algorithm approach is presented in Ref. [23]. Building upon the previous work, the authors of Ref. [24] demonstrate that by applying preprocessing the computational load on the quantum computer can be reduced and, in a generalization of the multitarget

search for single targets, the efficiency of the basic quantum search oracle for the MQ problem over the binary field can be improved. In Ref. [25], the authors present a quantum version of BOOLEANSOLVE [26], which is currently the fastest asymptotic algorithm for classically solving systems of nonlinear Boolean equations, that takes advantage of Grover's quantum algorithm. Note that Ref. [27] also proposed a new Gröbner-based quantum algorithm for solving quadratic equations with a complexity comparable to QUANTUMBOOLEANSOLVE (we refer to Ref. [25] for further details). Finally, in Ref. [28], the authors show how to reduce the use of quantum random access memory (RAM) and circuit complexity by delegating some precomputations to a classical computer.

Note that all the above quantum techniques have only considered the use of universal fault-tolerant quantum computers. Therefore the aforementioned methods cannot be implemented on current quantum computers without error correction. In contrast, the use of quantum annealers to solve systems of multivariate equations over a finite field is still unexplored, and in this paper, we try to fill this gap.

In this paper, we explore how quantum annealing can be used for solving multivariate systems of quadratic equations over binary fields, namely, the MQ problem. We present different methodologies to translate the MQ problem into a Hamiltonian that can be solved by a quantum annealer. To support our proposal, we provide results obtained by running examples on D-Wave's Advantage quantum device [13], a commercially available quantum computer. Our approach takes into consideration the decomposition of multiqubit terms up to at most two-qubit interactions, which is a constraint of the underlying architecture of near-term quantum annealing devices.

Our work highlights the main obstacle of mapping Boolean systems of equations into Hamiltonian ground states, which is the fact that the inherent correlations in operations over the binary field have to be encoded into a real Hamiltonian, at the cost of an overhead. We present two alternative methods to circumvent the exponential overhead in quantum memory that the naive transformation would incur. Moreover, we also introduce an iterative approach that aids quantum annealing devices in finding the ground state of the Hamiltonian by repeatedly shrinking the search space using information gained in previous executions of the system. This method allowed us to successfully solve small instances of the MQ problem using current D-Wave devices.

This paper is organized as follows. In Sec. II, we introduce the background required for this work. Next, in Sec. III we present our methodologies to translate a given MQ problem into a Hamiltonian in the context of quantum annealing and analyze the needed resources. Finally, in Secs. IV and V we present the results of our experiments with the D-Wave quantum annealer and the conclusions of this work, respectively.

II. PRELIMINARIES

We denote by $\mathbb{F}_q$ the finite field with q elements. $\mathbb{F}_q^n$ is the set of all vectors of length n, viewed as an $\mathbb{F}_q$-vector space. For compactness, we sometimes denote with $\vec{x}$ the vector $(x_1, \ldots, x_n)$.

The MQ problem is defined as follows. The input of the problem consists of m quadratic polynomials $p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n) \in \mathbb{F}_q[x_1, \ldots, x_n]$ in n variables $x_1, \ldots, x_n$ and coefficients in a finite field $\mathbb{F}_q$. The output of the problem is given by the set of $(a_1, \ldots, a_n) \in \mathbb{F}_q^n$ for which $p_i(a_1, \ldots, a_n) = 0$ for all $i = 1, \ldots, m$. The vector $(a_1, \ldots, a_n)$ is called a *solution* of the *system of equations*

$$p_i(x_1, \ldots, x_n) = 0, \quad i = 1, \ldots, m. \tag{1}$$

Three variants of the MQ problem can be defined. (1) The *decision* variant asks to determine if Eq. (1) has a solution. (2) The *search* variant asks to find a solution of Eq. (1), if there is one. (3) The *exhaust* variant asks to find all solutions of Eq. (1).

The decision variant of MQ is known to be an NP-complete problem [29]. It is easy to observe that solving the search variant also solves the decision one. On the other hand, by iteratively guessing each variable, it is possible to solve the search variant by solving the decision variant at most n times.

For practical purposes, one is usually interested in the search variant, and sometimes in the exhaust variant. From now on, unless stated otherwise, "MQ problem" refers to its search variant. Furthermore, we will focus on the Boolean case, i.e., where $q = 2$. In this case, polynomials are called Boolean polynomials, and the corresponding unique map f from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is called a Boolean function. It is common to refer to the Boolean polynomial as the algebraic normal form (ANF) of f, which we indicate with $f^{(\mathbb{F})}$. In this paper, we will also need another representation of f called the numerical normal form (NNF), which we indicate with $f^{(\mathbb{Z})}$.

Definition 1. Let f be a Boolean function on $\mathbb{F}_2^n$ taking values in the integer ring $\mathbb{Z}$. We call the *numerical normal form (NNF)* of f the following expression of f as a polynomial:

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}_2^n} \lambda_u \left(\prod_{i=1}^{n} x_i^{u_i} \right) = \sum_{u \in \mathbb{F}_2^n} \lambda_u X^u,$$

with $\lambda_u \in \mathbb{Z}$ and $u = (u_1, \ldots, u_n)$.

With abuse of notation and when clear from the context, we sometimes write $f = f^{(\mathbb{F})} = f^{(\mathbb{Z})}$, and we indicate with "+" the addition either over $\mathbb{F}_2$ or over another field or ring. Given a Boolean function, its ANF and NNF are unique. It is also worth noting that, in general, if a Boolean function in ANF has about k terms (i.e., nonzero coefficients), then its corresponding NNF will contain about 2^k terms (see Ref. [30] for detailed proof). As described in this paper, this significant increase in the number of terms turns out to be the main obstacle when trying to solve Boolean polynomial systems using annealing evolution. We finally refer to Ref. [31] for an exhaustive introduction to Boolean functions.

Example 1. An example of a quadratic Boolean polynomial system with $n = 4$ variables and $m = 4$ equations is given below:

$$\begin{aligned} x_1x_2 + x_1x_3 + x_1x_4 + x_1 + x_2x_3 + x_2x_4 + x_2 + x_3x_4 + x_4 &= 0, \\ x_1x_2 + x_1x_3 + x_2 + x_3x_4 + x_3 &= 0, \\ x_1x_2 + x_1x_3 + x_2x_3 + x_2 + x_3 + x_4 &= 0, \\ x_1x_3 + x_2x_4 + x_4 + 1 &= 0. \end{aligned}$$

The polynomials are given in algebraic normal form, and the only solutions of the system are the two binary vectors

(1, 0, 1, 0) and (0, 0, 1, 1). For example, the numerical normal form of $f^{(\mathbb{F})} = x_1x_3 + x_2x_4 + x_4 + 1$ (note that the addition is over $\mathbb{F}_2$) is given by $f^{(\mathbb{Z})} = -2x_1x_2x_3x_4 + 2x_1x_3x_4 - x_1x_3 + x_2x_4 - x_4 + 1$ (note that the addition is over the integer ring $\mathbb{Z}$).

III. FORMALIZING THE PROBLEM IN A QUANTUM ANNEALER

In the early 2000s, Farhi *et al.* proposed a new universal quantum computation model based on the quantum adiabatic theorem [1,32]. The so-called adiabatic quantum computation model was shown to be polynomially equivalent to the quantum gate-based model proposed by Deutsch in 1989 [33,34] and is one of the most promising models of quantum computing due to its natural robustness against errors [35]. The adiabatic theorem guarantees that if the Hamiltonian that dictates the energy of a quantum system is modified slowly enough, a quantum state will remain in its instantaneous ground state during the evolution [36,37]. This implies that we can encode the solution of a hard problem, the MQ problem in this case, into the ground state of a problem Hamiltonian H_p and then, starting from an easy-to-prepare ground state of an initial Hamiltonian H_0, drive the system slowly to the problem Hamiltonian to then measure its solution.

A less restrictive, more hardware-friendly, technique to solve classical problems is quantum annealing. Quantum annealing also follows the evolution of a quantum Hamiltonian in order to find low-energy configurations of the system but does not demand adiabatic evolution and forgoes universality. Devices such as the D-Wave quantum annealer, while not being universal quantum computers due to their limitations, are still useful for solving hard optimization problems [38,39]. Hard classical problems that can be codified to a problem Hamiltonian by only using the computational basis of the system, as is the case for the MQ problem tackled in this paper, are ideal for these available quantum annealers.

We want to solve the MQ problem defined in Sec. II, where we are given a set of m quadratic polynomials $p_1(x_1, \ldots x_n), \ldots, p_m(x_1, \ldots x_n)$ over the binary field and we are tasked with finding $\vec{x}$ so that all $\vec{p}$ are equal to zero, via quantum annealing. Therefore we need to create a Hamiltonian with a ground state that encodes the solution to this problem.

A. Direct embedding

A first direct approach is penalizing with positive energy each of the equations $p_i(\vec{x})$ that is not fulfilled. The corresponding problem Hamiltonian can be constructed as

$$H_p = \sum_{i=1}^{m} p_i(\vec{x}), \tag{2}$$

as it contributes with positive energy if the input bits for $p_i(\vec{x})$ do not result in a zero solution.

Usually, the polynomials $p_i(\vec{x})$ in Eq. (2) are given in ANF since bitwise operations are performed over the binary field $\mathbb{F}_2$. However, the quantum Hamiltonian we can encode into a quantum annealer device does not function with binary algebra; each positive term only adds more energy to the final state. Therefore each polynomial $p_i(\vec{x})$ has to be given in its NNF. This transformation can be obtained by recursively applying the change

$$(x_i + x_j) \longrightarrow x_i + x_j - 2x_i \cdot x_j \tag{3}$$

to the original ANF equations, where, with abuse of notation, the symbol $+$ on the left is the addition over $\mathbb{F}_2$, while the symbols $+$ and $-$ on the right are the regular addition and subtraction over the integer ring. This transformation introduces multiqubit interaction terms (i.e., terms of degree greater than 1) that were not present in the ANF of $p_i(\vec{x})$. In general, all combination of monomials present in the ANF of $p_i(\vec{x})$ will appear in the NNF. Keep in mind that, in a binary field, it holds that $x^2 = x$ (since the values of the variables x_i are either 0 or 1); there are no powers in the monomials of the ANF or the NNF.

This transformation gives rise to a different issue: The chip architecture of currently available quantum annealers only allows for two-qubit interactions. Thus, to run a quantum annealing protocol on a real device, the interactions of the Hamiltonian have to be reduced. For a general many-body Hamiltonian, its interactions can be reduced to two-body interactions using perturbation theory by adding ancilla qubits [40–42]. If all the problem Hamiltonian parts share the same basis, as is the case for a classical Hamiltonian such as ours, the reduction can be performed without perturbation theory [43,44]. This reduction method yields a new Hamiltonian with a different energy spectrum but equal ground state and energy, therefore not altering the solution of the problem, and is the one we follow for the direct embedding.

The method consists of exchanging a two-qubit interaction for an ancilla, reducing by 1 the order of the interaction. A penalty function is then introduced into the Hamiltonian that adds energy when the value of the ancilla is not equal to the product of the original two qubits. The penalty function can be written as

$$s(x_i, x_j, x_{ij}) = 3x_{ij} + x_ix_j - 2x_ix_{ij} - 2x_jx_{ij}, \tag{4}$$

where x_{ij} is the label given to the ancillary qubit that is substituted. It can be seen that $s(x_i, x_j, x_{ij}) = 0$ if $x_ix_j = x_{ij}$ and $s(x_i, x_j, x_{ij}) \geqslant 1$ otherwise. This keeps the ground state and energy unchanged.

Furthermore, a single ancilla x_{ij} can be used for all terms in the Hamiltonian, where the term x_ix_j appears. This is achieved by applying the substitution

$$\sum_K \alpha_{ijK} x_i x_j x_K \longrightarrow \sum_K (\alpha_{ijK} x_{ij} x_K + (1 + |\alpha_{ijK}|) s(x_i, x_j, x_{ij})), \tag{5}$$

where the index K is the product of multiple other variables in all terms where x_ix_j is present. It can be shown that this transformation also yields a Hamiltonian with the same ground state [44].

When this procedure is used to reduce large multiqubit terms, the resulting final Hamiltonian will have large coefficients. This introduces a problem for real-life implementation since the machine precision for coefficients of quantum annealing devices such as D-Wave's is limited. The quantum

annealer developed by D-Wave scales the given coefficients between $[-1, 1]$ when introducing them to the machine, so small coefficients can vanish when translated into weights in the presence of other large parameters.

An alternative transformation is proposed in Ref. [44] with the aim of reducing the precision needed for the control of the device. Introducing the term

$$\delta_{ij} = \max\left(\sum_{K,\alpha_{ijK}>0} \alpha_{ijK}, \sum_{K,\alpha_{ijK}<0} -\alpha_{ijK}\right), \tag{6}$$

the substitution given in Eq. (5) can be rewritten as

$$\sum_K \alpha_{ijK} x_i x_j x_K \longrightarrow \sum_K \alpha_{ijK} x_{ij} x_K + (1+\delta_{ij}) s(x_i, x_j, x_{ij}), \tag{7}$$

while still keeping the desired ground state. This reduces, but does not completely solve, the precision problem.

If a given n-qubit Hamiltonian contains multiqubit interactions involving all of its constituents, that is, an n-qubit Hamiltonian with up to n-body terms, one would require $2^{\frac{n+2}{2}} - 2$ total qubits to reduce all possible combinations of qubit interactions to two-body terms for an even n ($3 \times 2^{\frac{n-1}{2}} - 2$ for odd n). This can be achieved by dividing the total qubit register into two fully connected graphs using ancillary variables and connecting both graphs with another ancillary qubit. Unfortunately, this will be the case for a general conversion from ANF to NNF due to the fact that an n-term sum in ANF will generally require

$$\sum_{k=1}^{n} \binom{n}{k} = 2^n - 1 \tag{8}$$

terms for the equivalent NNF equation. Therefore one would need an exponential amount of quantum resources, ancillary qubits in this case, to encode the ground state into a Hamiltonian following this first direct approach.

B. Truncated embedding

This problem can be circumvented by partitioning the original polynomials $p_i(\vec{x})$ into smaller pieces with k-bounded length using ancillary variables. It is straightforward to see that a sum of n_i monomials can be reduced to sums of up to k terms by adding ancillae in the form

$$\begin{aligned} x_1 + \cdots + x_{n_i} = 0 &\rightarrow x_1 + \cdots + x_{k-1} + a_1 = 0 \\ &\rightarrow a_1 + x_k + \cdots + x_{2k-2} + a_2 = 0 \\ &\cdots \\ &\rightarrow a_l + x_{n_i-k+1} \cdots + x_{n_i} = 0, \end{aligned} \tag{9}$$

at the cost of expanding the number of equations to $\frac{n_i-2}{k-2}$ using $l = \frac{n_i-2}{k-2} - 1 = \frac{n_i-k}{k-2}$ ancilla variables we have labeled a_i. A similar technique can also be used when encoding the MQ problem as a Boolean satisfiability problem (SAT) instance [45].

To have more precise control of the total number of ancillary qubits added to decompose the multiqubit terms, we need to ensure that the parameter k is also the maximum number of multibody interactions. For that reason, we first introduce ancilla variables to substitute the two-qubit terms in the original ANF representation adding the required penalty functions. In the worst-case scenario, where all combinations of two-body interactions appear, we will need to add $\binom{n}{2} = n(n-1)/2$ ancillary variables. This, however, does not change the total number of monomials in the truncated system of equations.

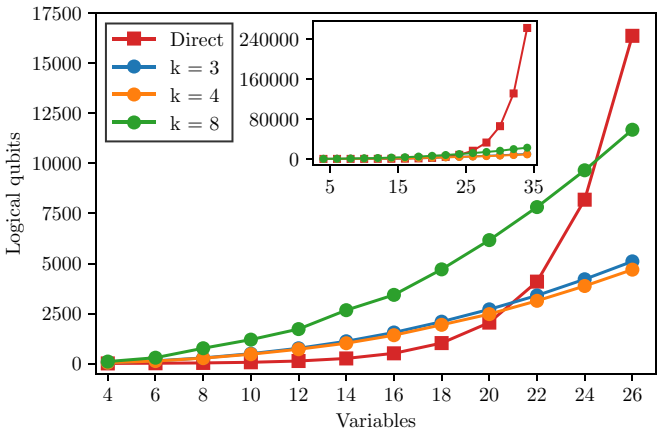

FIG. 1. Number of logical qubits needed to embed an MQ problem into the ground state of a Hamiltonian for the direct and truncated approaches. It can be seen how the direct approach, while more resource efficient in small systems, quickly outpaces the truncated approach. The inset shows the scaling for a larger number of variables. Note also that the optimal value for the cutoff variable is $k = 4$.

The maximum number of qubits needed to represent an MQ problem with m equations involving n variables as a two-body Hamiltonian will then be

$$\sum_{i=1}^{m} \left[\frac{n_i - 2}{k-2}(2^{\frac{k+2}{2}} - 2 - k) + \frac{n_i - k}{k-2}\right] + \binom{n}{2} + n, \tag{10}$$

where n_i is the number of monomials in each $p_i(\vec{x})$ of Eq. (1) and k (even) is the length of the partitions.

The differences in scaling between the truncated approach with different values for k and the direct embedding can be seen in Fig. 1. While the direct approach is more efficient when the number of variables is small, its exponential scaling quickly makes it unfeasible when compared with the truncated approach. For the polynomial approach, the cutoff variable k defines its scaling. We note that the scaling is optimal for $k = 4$. Additionally, the precision issues raised in the substitution scheme will be significantly attenuated in the truncated embedding since now the nonlocality of the ancillae is governed by k and not the total number of variables.

A partition length of $k = 4$ minimizes the total number of ancillae since the exponential term $2^{\frac{k+2}{2}}$ dominates and the truncation of the original equation means that half the number of parameters are needed than are needed for $k = 3$. Precisely, a four-term sum will only need two extra ancillary variables to reduce it to up to two-body terms. Fixing the value for k, the total number of qubits needed reads

$$\frac{n^2}{2} + \frac{n}{2} - 4m + \frac{3}{2}\sum_{i=1}^{m} n_i. \tag{11}$$

TABLE I. Summary of Boolean operations and their penalty function implementation in a quantum annealer as only constant, single-qubit, and two-qubit monomials appear in MQ problems. The result is saved in the qubit corresponding to the variable z, while the variable x corresponds to other qubits involved in the Boolean operation. The subscripts c, t, and a correspond to *control*, *target*, and *ancilla*, respectively.

Gate	Boolean operation	Penalty function
NOT	$z = \bar{x}$	$2xz - x - z + 1$
Controlled-NOT	$z = x_c x_t$	$2x_c x_t - 2(x_c + x_t)z - 4(x_c + x_t)x_a + 4zx_a + x_c + x_t + z + 4x_a$
Toffoli	$z = x_{c1}x_{c2}x_t$	$-4x_{a1}x_{a2} + 4x_{a1}z - 4x_{a1}x_t - 2x_{a1}x_{c1} - 2x_{a2}x_{c2}2x_{a2}z + 2x_{a2}x_t + x_{c1}x_{c2} - 2x_t z + 4x_{a1} + 4x_{a2} + z + x_t$

We encounter now a polynomial scaling with the number of parameters under the condition that the total number of terms in the system of equations scales reasonably with the number of variables.

In order to obtain a qubit scaling that only depends on the number of variables n, we can use average values for both m and n_i. Generally, we will encounter as many equations as variables in the system, $m = n$, each with an average number of monomials given by the total possible combinations of terms with two-body interactions, $n_i \sim (n + \binom{n}{2})/2 = (n + n^2)/4$. These two approximations yield the new scaling

$$\tfrac{3}{8}n^3 + \tfrac{7}{8}n^2 - \tfrac{7}{8}n, \tag{12}$$

a polynomial of degree 3 in the number of variables of the problem.

C. Penalty embedding

An alternative way to embed the ground state of the MQ problem is to model the equations in their ANF using logical quantum gates such as CNOT or Toffoli gates which natively act over the $\mathbb{F}_2$ field and then reproduce that circuit as an adiabatic evolution using penalty functions. To be more precise, we model the MQ problem equations as Boolean operations on an output quantum register; that is, the actions of $+x_i$ and $+x_i x_j$ can be modeled to CNOT and Toffoli gates targeting the output qubit and controlled by qubits $\{x_i\}$ and $\{x_i, x_j\}$, respectively. Then a Hamiltonian is constructed with a ground state that follows the correct gate-by-gate implementation of the resulting circuit.

This method of circuit-to-Hamiltonian encoding using penalty functions is reminiscent of Feynman's Hamiltonian clock [46], where in order to create a Hamiltonian that faithfully represents the actions of a logical quantum circuit one would use an extra *clock register* where the time step of each applied quantum gate is stored. In this implementation, an ancillary *output* qubit register is added, which stores the result of the output qubit after each gate application.

The penalty functions needed to map the solution of an MQ problem into the ground state of a Hamiltonian are displayed in Table I. The *output* ancilla qubit z used in the penalty function of a given quantum gate will be used as the *target* qubit x_t in the penalty function of the immediately following gate. These penalty functions contribute with positive energy if the state of the qubits involved does not match the logical Boolean operation that they map. Additionally, the qubits used to initialize the *output* ancilla register are penalized if they are in the $|1\rangle$ state as we assume an initial state of the output qubit of $|0\rangle$. The same thing is applied to the *output* ancillae where the final result of applying each equation $p_i(\vec{x})$ in Eq. (1) is stored as we are interested in the solution where the output is zero.

It is straightforward to see that the quantum resources needed to apply this implementation are governed by the number of monomials present in the equations of a given MQ problem, as they will dictate the number of gates that are to be implemented. As discussed in Sec. III B above, the average number of monomials appearing in a given problem will scale as $O(n^3)$, and the ancilla overhead needed for the implementation of each CNOT or Toffoli gate, an extra one or two ancillae, respectively, will not change the overall scaling. Therefore, up to the particularities of each implementation, both the truncated and the penalty function embedding will scale similarly, and in large system sizes outclass the direct embedding.

However, it is crucial to mention the number of physical qubits needed for the implementation when assessing the actual quantum resources. Due to chip architecture constraints, mapping a Hamiltonian into a real quantum annealing device will require an overhead to account for nonlocal interactions. The D-Wave application programming interface (API) provides the automatic solver MINORMINER [47] to find a good embedding into their architecture. Highly nonlocal Hamiltonians will require a large number of physical qubits in order to represent each logical variable. Moreover, the amount of required physical qubits can change the scaling of a particular method, giving an edge to a more local embedding with more logical variables.

We show in Table II the comparison between both the truncated and penalty embeddings in terms of physical and logical qubits required for their implementation into the Advantage D-Wave machine. For different instances up to 12 logical variables, we show the amount of required physical quantum resources for both the truncated and the penalty embedding. Each embedding has been averaged over ten instances in order to reduce the uncertainty due to the minimization method provided by D-Wave. We note that both the truncated

TABLE II. Number of logical and physical qubits needed to map the truncated and penalty embedding for different numbers of variables. Physical qubit values are averaged over ten instances of the MINORMINER algorithm provided by the D-Wave API [47].

Variables	4	6	8	10	12
Truncated (logical)	30	90	231	451	718
Truncated (physical)	55.6	223.0	758.0	1627.8	2645.2
Penalty (logical)	61	150	345	645	1005
Penalty (physical)	105.1	309.4	864.1	1940.6	3436.5

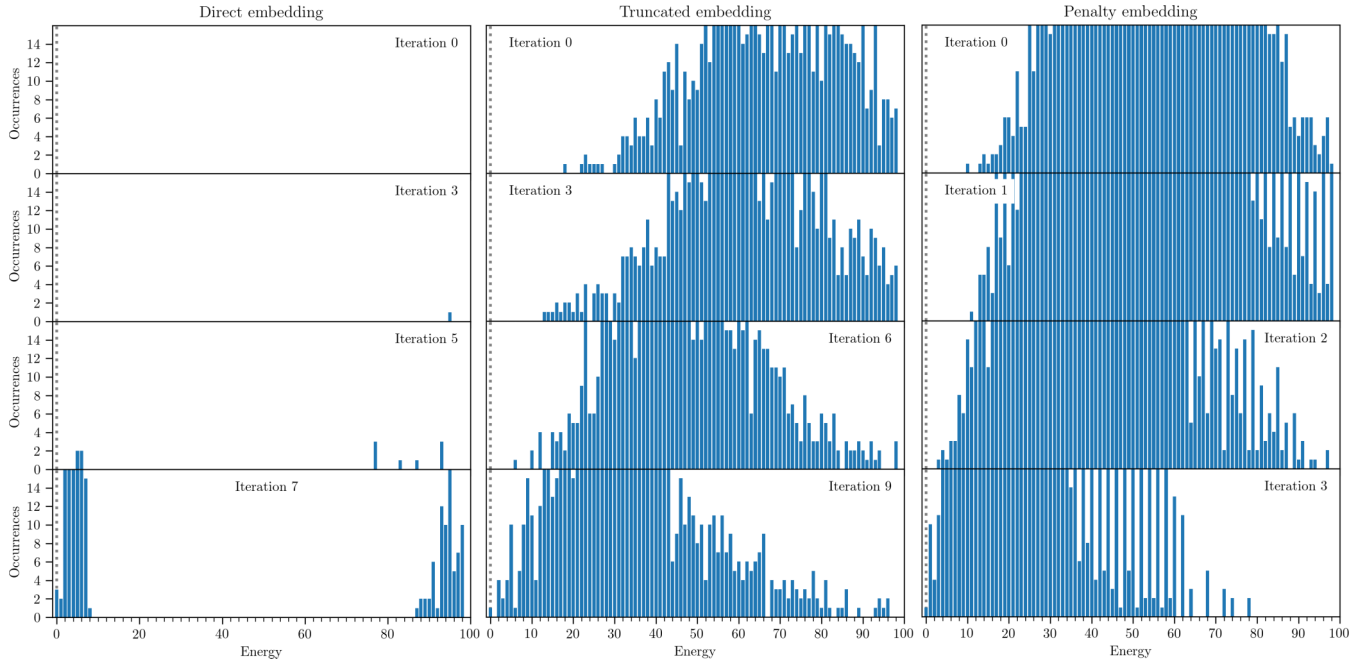

FIG. 2. Samples below 100 arb. units of energy after a 1000-sample execution on D-Wave's Advantage machine for the different MQ problem embeddings presented where the ground state was first observed. We have implemented problems with a similar number of physical qubits required in the first embedding of the problem. The direct embedding (left) encodes a nine-variable problem in 46 logical qubits that are mapped to 179 physical ones, the truncated (center) and penalty (right) embeddings both encode a five-variable problem, with 67 logical qubits and 167 physical qubits for the truncated embedding and 114 logical qubits and 221 physical qubits for the penalty embedding. We show four different iterations in our iterative method to highlight how the energy approaches the ground state as the system gets smaller both in logical and physical qubits. The ground-state energy of this problem is 0, depicted as a gray dotted line.

embedding and the penalty embedding scale in a similar manner when mapped into the physical qubits of a given chip architecture with a worse, albeit still polynomial, overall scaling.

IV. RESULTS

In this section, we encode some reduced MQ problem instances using the methods presented in Sec. III in order to be solved using D-Wave machines. We also propose an iterative method to aid in finding a singular correct solution in large, highly correlated, systems.

So far we have presented several ways to encode the solution of an MQ problem into the ground state of a Hamiltonian that can be used for quantum annealing. The implementation of such a protocol on a real quantum device, however, will require adjusting to the specifics of the particular machine. The problem Hamiltonian assumes the implementation of all-to-all interaction. However, this is unrealistic because the superconducting chips for quantum annealing provided by D-Wave's Advantage device support a Pegasus chip architecture [48] and therefore the qubits need to be mapped in accordance with that restriction. The solution to this problem is the introduction of *qubit chains*.

A logical qubit will be extended into a chain of qubits when mapped into the physical chip of the quantum device. This means that different physical qubits, which will represent the same variable, are bound together by an interaction term, a *chain strength*, that penalizes members of the same qubit chain for being in different quantum states. We leave the mapping of the original variables to physical objects to the built-in compiler provided by the D-Wave library [47] and adjust the chain strength hyperparameter in order to not overpower the variables of the problem while measuring as few broken chains as possible. This mapping will result in a more complex evolution, and consequently poorer results, especially for Hamiltonians with a large number of nonlocal qubit interactions.

We present the results of running the Hamiltonians proposed in the different embedding schemes. The uppermost graphs in Fig. 2 show the results of sampling the final state of a quantum annealing evolved under each corresponding Hamiltonian for the direct, penalty, and truncated embedding, respectively. We state for each case the number of logical and physical qubits the problem needed to be mapped to. We decided to focus on a similar number of physical qubits needed; therefore the direct embedding was able to reach a nine-variable problem, while the truncated and penalty embeddings are limited to five variables. We note that with a small number of variables, the annealing process does not yield the exact ground state that encodes the solution of the problem, the quantum state with zero energy. Longer annealing times, more precise control of the annealing schedule, or higher-quality qubits are ways to improve the results. However, current quantum annealers might not have the capabilities of tuning those parameters to the required specifications of large problems. In order to achieve the ground-state energy of the problem in a machine-agnostic way, we propose an iterative algorithm that closes in on a smaller, easier-to-solve, subspace where the ground state might be located.

As detailed in Sec. III, the proliferation of ancillary qubits in the different proposed embedding options appears when reducing the multiqubit interactions into at most two-qubit interaction terms. This means that most of these added ancillae will represent products of other variables and will therefore have a stronger penalization than the original variables that they represent. That is, the wrong state of certain ancillary variables is penalized with a higher amount of energy than others. The following is a heuristic iterative method where we use that to our advantage.

After running an annealing protocol on a quantum device, if no quantum state with zero energy has been found, we may look at some of the low-energy configurations of the obtained samples. If some qubits are found in the same result in all of the lowest-energy states, we can assume that the Hamiltonian penalizes those variables more than the others. We can narrow the subsequent search space by substituting that variable in the original Hamiltonian by its, now known, preferred value. The more the search space is reduced, the easier it is for the quantum annealing device to find the lowest-energy solution.

The amount of low-energy solutions to check for the same value of the ancilla variables is a hyperparameter that can be optimized. On the one hand, if we set the value too low, we might be excluding the ground state from the reduced search space by fixing ancillae to a wrong outcome. On the other hand, if we set it too high, we might not find any variable that lays in the same output for all low-energy configurations. More sampling at each iteration will also enhance our ability to fix ancillae but will impact the overall run time of the algorithm. The number of fixed parameters per run will depend on the problem, encoding, and quality of the quantum device. We used a heuristic approach when tuning the number of low-energy solutions checked. A method to check whether the reduced subspace no longer contains the solution can be devised. If the lowest-energy sample of the reduced Hamiltonian is lower than its equivalent value in the original Hamiltonian, adding back up the fixed variables of the original setting, then we have excluded the original ground state from the reduced subspace. The different rows in Fig. 2 show how this iterative approach indeed helps in finding the ground state of the problem. As more iterations go by and more ancillae are fixed, the system starts finding lower-energy solutions until the ground state with zero energy is reached.

The first row in Fig. 2 shows the initial run of the algorithm. Then the following two rows are some snapshots of the energy samples during the iterative algorithm, and the last row showcases the first iteration where a state with zero energy is reached. The direct embedding shows the result of having a highly nonlinear Hamiltonian with large parameters. The initial runs are very far away from the ground state, and it is not until the more volatile variables are fixed that the ground state can be found. The truncated and penalty embeddings behave in a similar way to each other. It can be seen how after each iteration the median energy gets closer and closer to the ground energy until it is reached. We note that the penalty embedding, in spite of requiring more qubits to embed the problem Hamiltonian, reaches the ground state with fewer iterations. This can be attributed to the lower coefficients that are needed to map the problem, making it more suited to an annealer machine such as the one provided by D-Wave.

V. CONCLUSION

Our work is a first step towards demonstrating the efficiency of quantum annealing computations in solving the MQ problem, using practical experiments on the existing D-Wave quantum annealing platform. We show that we can construct a Hamiltonian with a ground state encoding the solution of the problem and subsequently find it using a quantum annealer. We propose different methods for the embedding of the problem into a Hamiltonian using a polynomial amount of quantum resources. As quantum technology advances, we foresee that the evolution of quantum annealing architectures (e.g., support to n-body interactions or larger coherence times) might provide a quantum advantage when solving such problems as the required numbers of ancilla qubits and required quantum control would decrease.

We have introduced an algorithm that simplifies the problems by fixing ancillary qubits that are easy to find for the quantum device in order to more reliably find the ground state of the more complex qubits with finer parameters. This method can help when dealing with large amounts of qubits in near-term devices and could be applied in problems beyond the scope of what is studied in this paper.

As an estimate of the quantum resources needed to solve state-of-the-art MQ problems, we refer to the Fukuoka MQ Challenge website [49], where the largest unsolved instances of MQ problems can be found. The type I challenge problem instance of 37 equations and 74 variables would require an estimate of under 80 000 logical qubits for its solution to be mapped into the ground state of a two-body Hamiltonian. For the type IV challenge with 69 equations and 105 variables, one would require under 300 000 logical qubits for the embedding. Quantum annealers are still far away from being able to tackle the problems at the edge of what is classically solvable; however, quantum technologies are still emerging, and new devices with more, and higher-quality, qubits are being currently developed.

For a detailed implementation, we refer to our code made available on GitHub [50].

ACKNOWLEDGMENTS

The authors would like to thank Andre Esser for insights. M.M. is partially supported by the TRUSTIND project, under Grant Agreement No. KK-2020/00054, from the Department of Economic Development and Infrastructures of the Basque Government. M.M. is also a member of the Intelligent Systems for Industrial Systems research group of Mondragon Unibertsitatea (IT1676-22), supported by the Department of Education, Universities and Research of the Basque Country.

[1] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv:quant-ph/0001106.

[2] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, Quantum annealing: A new method for minimizing multidimensional functions, Chem. Phys. Lett. **219**, 343 (1994).

[3] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, Quantum annealing of a disordered magnet, Science **284**, 779 (1999).

[4] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom *et al.*, Quantum annealing with manufactured spins, Nature (London) **473**, 194 (2011).

[5] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, Finding low-energy conformations of lattice protein models by quantum annealing, Sci. Rep. **2**, 571 (2012).

[6] S. Mandra, Z. Zhu, W. Wang, A. Perdomo-Ortiz, and H. G. Katzgraber, Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches, Phys. Rev. A **94**, 022337 (2016).

[7] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models, Phys. Rev. X **7**, 041052 (2017).

[8] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, and M. H. Amin, Quantum variational autoencoder, Quantum Sci. Technol. **4**, 014001 (2018).

[9] M. Benedetti, J. Realpe-Gómez, and A. Perdomo-Ortiz, Quantum-assisted Helmholtz machines: A quantum–classical deep learning framework for industrial datasets in near-term devices, Quantum Sci. Technol. **3**, 034007 (2018).

[10] Y. Ding, J. Gonzalez-Conde, L. Lamata, J. D. Martín-Guerrero, E. Lizaso, S. Mugel, X. Chen, R. Orús, E. Solano, and M. Sanz, Towards prediction of financial crashes with a D-Wave quantum computer, arXiv:1904.05808.

[11] A. Perdomo-Ortiz, A. Feldman, A. Ozaeta, S. V. Isakov, Z. Zhu, B. O'Gorman, H. G. Katzgraber, A. Diedrich, H. Neven, J. de Kleer, B. Lackey, and R. Biswas, Readiness of Quantum Optimization Machines for Industrial Applications, Phys. Rev. Appl. **12**, 014004 (2019).

[12] M. Wilson, T. Vandal, T. Hogg, and E. G. Rieffel, Quantum-assisted associative adversarial network: Applying quantum annealing in deep learning, Quantum Mach. Intell. **3**, 19 (2021).

[13] https://www.dwavesys.com.

[14] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., New York, USA, 1990).

[15] R. Impagliazzo and R. Paturi, On the complexity of k-SAT, J. Comput. Syst. Sci. **62**, 367 (2001).

[16] G. Bard, *Algebraic Cryptanalysis* (Springer, New York, 2009).

[17] J. Ding, M.-S. Chen, M. Kannwischer, J. Patarin, A. Petzoldt, D. Schmidt, and B.-Y. Yang, Name of proposal: Rainbow, available in the supporting documentation of Rainbow submission package to NIST, https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Rainbow-Round3.zip.

[18] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem, GeMSS: A Great Multivariate Short Signature, available in the supporting documentation of GeMMS submission package to NIST, https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/GeMSS-Round3.zip.

[19] G. V. Bard, Algorithms for solving linear and polynomial systems of equations over finite fields with applications to cryptanalysis, Ph.D. thesis, University of Maryland, 2007.

[20] C. Mou, Solving polynomial systems over finite fields: Algorithms, implementation and applications, Ph.D. thesis, Université Pierre et Marie Curie, 2013.

[21] E. Ullah, New techniques for polynomial system solving, Ph.D. thesis, Universität Passau, 2012.

[22] C. Eder and J.-C. Faugère, A survey on signature-based algorithms for computing Gröbner bases, J. Symb. Comput. **80**, 719 (2017).

[23] P. Schwabe and B. Westerbaan, Solving binary $\mathcal{MQ}$ with Grover's algorithm, in *International Conference on Security, Privacy, and Applied Cryptography Engineering* (Springer, New York, 2016), pp. 303–322.

[24] B. Pring, Exploiting preprocessing for quantum search to break parameters for $\mathcal{MQ}$ cryptosystems, in *International Workshop on the Arithmetic of Finite Fields* (Springer, New York, 2018), pp. 291–307.

[25] J.-C. Faugere, K. Horan, D. Kahrobaei, M. Kaplan, E. Kashefi, and L. Perret, Fast quantum algorithm for solving multivariate quadratic equations, arXiv:1712.07211.

[26] M. Bardet, J.-C. Faugère, B. Salvy, and P.-J. Spaenlehauer, On the complexity of solving quadratic Boolean systems, J. Complex. **29**, 53 (2013).

[27] D. J. Bernstein and B.-Y. Yang, Asymptotically faster quantum algorithms to solve multivariate quadratic equations, in *International Conference on Post-Quantum Cryptography* (Springer, New York, 2018), pp. 487–506.

[28] J.-F. Biasse and B. Pring, A framework for reducing the overhead of the quantum oracle for use with Grover's algorithm with applications to cryptanalysis of SIKE, J. Math. Cryptol. **15**, 143 (2021).

[29] A. S. Fraenkel and Y. Yesha, Complexity of problems in games, graphs and algebraic equations, Discrete Appl. Math. **1**, 15 (1979).

[30] E. Bellini and M. Sala, A deterministic algorithm for the distance and weight distribution of binary nonlinear codes, Int. J. Inf. Coding Theory **5**, 18 (2018).

[31] C. Carlet, Boolean functions for cryptography and error-correcting codes, in *Boolean Models and Methods in Mathematics, Computer Science, and Engineering* (Cambridge University Press, Cambridge, UK, 2010), Chap. 8, pp. 257–397.

[32] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, Science **292**, 472 (2001).

[33] D. E. Deutsch, Quantum computational networks, Proc. R. Soc. London Ser. A **425**, 73 (1989).

[34] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, Adiabatic quantum computation is equivalent to standard quantum computation, SIAM Rev. **50**, 755 (2008).

[35] A. M. Childs, E. Farhi, and J. Preskill, Robustness of adiabatic quantum computation, Phys. Rev. A **65**, 012322 (2001).

[36] M. Born and V. Fock, Beweis des Adiabatensatzes, Z. Phys. **51**, 165 (1928).

[37] T. Kato, On the adiabatic theorem of quantum mechanics, J. Phys. Soc. Jpn. **5**, 435 (1950).

[38] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E **58**, 5355 (1998).

[39] A. Das and B. K. Chakrabarti, Colloquium: Quantum annealing and analog quantum computation, Rev. Mod. Phys. **80**, 1061 (2008).

[40] S. Bravyi, D. P. DiVincenzo, D. Loss, and B. M. Terhal, Quantum Simulation of Many-Body Hamiltonians Using Perturbation Theory with Bounded-Strength Interactions, Phys. Rev. Lett. **101**, 070503 (2008).

[41] S. P. Jordan and E. Farhi, Perturbative gadgets at arbitrary orders, Phys. Rev. A **77**, 062329 (2008).

[42] Y. Cao, R. Babbush, J. Biamonte, and S. Kais, Hamiltonian gadgets with reduced resource requirements, Phys. Rev. A **91**, 012315 (2015).

[43] J. D. Biamonte, Nonperturbative k-body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins, Phys. Rev. A **77**, 052331 (2008).

[44] R. Babbush, B. O'Gorman, and A. Aspuru-Guzik, Resource efficient gadgets for compiling adiabatic quantum optimization problems, Ann. Phys. (Berlin) **525**, 877 (2013).

[45] G. V. Bard, N. T. Courtois, and C. Jefferson, Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over GF(2) via SAT-solvers, Cryptology ePrint Archive, Report No. 2007/024, 2007.

[46] R. P. Feynman, Quantum mechanical computers, Opt. News **11**, 11 (1985).

[47] https://docs.ocean.dwavesys.com/projects/minorminer/en/latest/.

[48] N. Dattani, S. Szalay, and N. Chancellor, Pegasus: The second connectivity graph for large-scale quantum annealing hardware, arXiv:1901.07636.

[49] T. Yasuda, Fukuoka MQ Challenge, available at https://www.mqchallenge.org/.

[50] S. Ramos-Calderer and R. Lin, https://github.com/qiboteam/mq-problem-quantum-annealing, 2021.