Soft Computing
https://doi.org/10.1007/s00500-018-3034-z

FOCUS

Biclustering with a quantum annealer

Lorenzo Bottarelli[1] · Manuele Bicego[1] · Matteo Denitto[1] · Alessandra Di Pierro[1] · Alessandro Farinelli[1] · Riccardo Mengoni[1]

Abstract

Several problem in Artificial Intelligence and Pattern Recognition are computationally intractable due to their inherent complexity and the exponential size of the solution space. One example of such problems is biclustering, a specific clustering problem where rows and columns of a data-matrix must be clustered simultaneously. Quantum information processing could provide a viable alternative to combat such a complexity. A notable work in this direction is the recent development of the D-Wave computer, whose processor has been designed to the purpose of solving Quadratic Unconstrained Binary Optimization (QUBO) problems. In this paper, we investigate the use of quantum annealing by providing the first QUBO model for biclustering and a theoretical analysis of its properties (correctness and complexity). We empirically evaluated the accuracy of the model on a synthetic data-set and then performed experiments on a D-Wave machine discussing its practical applicability and embedding properties.

Keywords Quantum annealing · D-Wave · Biclustering

1 Introduction

Biclustering, also known in other scenarios as subspace clustering, is a term used to encompass a large set of data mining techniques generally aimed at "performing simultaneous row-column clustering" of a data matrix (Madeira and Oliveira 2004). It is used in several different scenarios, such as document analysis (Dhillon 2001), market segmentation (Dolnicar et al. 2012), recommender systems (Mukhopadhyay et al. 2014) and, most importantly, expression microarray data analysis (Oghabian et al. 2014; Madeira and Oliveira 2004; Badea 2009; Prelić et al. 2006; Flores et al. 2013). In this last scenario, the starting point is a matrix whose rows and columns represent genes and experiments, respectively. Each entry of the matrix measures the expression level of a gene in a specific experiment. Biclustering aims to find clusters of genes which show a coherent behavior in subsets of experiments. This permits the discovery of co-regulation mechanisms. Answering this task can provide invaluable information to biologists, given the ever increasing amount of data that they have to analyze.

Different biclustering techniques have been proposed in the past (Cheng and Church 2000; Ayadi et al. 2012; Tu et al. 2011; Bicego et al. 2010; Denitto et al. 2014), each one characterized by different features, such as computational complexity, effectiveness, interpretability and optimization criterion—cf. (Madeira and Oliveira 2004; Prelić et al. 2006; Flores et al. 2013; Henriques et al. 2015; Henriques and Madeira 2014) for a general review. Some of these approaches aim at adapting a given clustering technique to the biclustering problem, for example by repeatedly performing rows and columns clustering. However, the majority of recent works aim at proposing novel models for biclustering, where rows and columns are analyzed simultaneously

Communicated by C. M. Vide, A. H. Dediu.

✉ Lorenzo Bottarelli
lorenzo.bottarelli@univr.it

Manuele Bicego
manuele.bicego@univr.it

Matteo Denitto
matteo.denitto@univr.it

Alessandra Di Pierro
alessandra.dipierro@univr.it

Alessandro Farinelli
alessandro.farinelli@univr.it

Riccardo Mengoni
riccardo.mengoni@univr.it

[1] Department of Computer Science, University of Verona, Verona, Italy

Published online: 29 January 2018

(as opposed to clustering rows and columns separately) (Tu et al. 2011). This has several advantages for what concerns the performance of the biclustering process that is significantly more accurate. However, such accuracy comes at a price as such models typically involve a large amount of variables and relationships. Specifically, the typical candidate data for biclustering are represented by a matrix with thousands of column/rows (Madeira and Oliveira 2004). Moreover, the underlying optimization task required by the model is NP-hard leading to severe restrictions on the practical applicability of those approaches. In order to combat such complexity, recent works typically relax the model or use heuristic, greedy approaches, hence giving up optimality of the solution.

In this paper, we investigate the applicability of a meta-heuristic, called Quantum Annealing (QA) (Finnila et al. 1994; Kadowaki and Nishimori 1998; Santoro and Tosatti 2006), to the global optimization problems underlying biclustering, by following some recent developments in the construction of quantum devices that physically realize quantum annealing. Similarly to the classical Simulated Annealing, QA is an optimization meta-heuristic that seeks the global optimum of an objective function by following a process inspired by the thermodynamic process of annealing. In this search, QA employs quantum fluctuations in order to escape local minima, i.e., it uses some quantum effects that allows the tunneling through narrow barriers separating local minima, rather than climbing over them as done classically by using thermal fluctuations. Apart from the recent theoretical demonstrations, this has also been demonstrated experimentally (Denchev et al. 2016). A fundamental contribution in this direction is due to D-Wave Systems Inc., which has commercialized some analog quantum devices designed to use quantum annealing to solve quadratic optimization problems.

Various works investigated the possibility of addressing typical Artificial Intelligence (AI) and Pattern Recognition (PR) problems by using QA. Examples include image recognition (Neven et al. 2008), Bayesian network structure learning (O’Gorman et al. 2015) and hard operational planning problems (Rieffel et al. 2015). As done in Rieffel et al. (2015) or in Neven et al. (2008) for image recognition, we show here an encoding of biclustering as a Quadratic Unconstrained Binary Optimization (QUBO) problem (Kochenberger et al. 2014), i.e., as a problem where the aim is to find an assignment for binary variables so as to minimize a quadratic objective function. The QUBO format corresponds to the input format required for the D-Wave superconducting adiabatic quantum computing processors. To the best of our knowledge this is the first study in this direction. A sampling algorithm for clustering was proposed in Kurihara et al. (2009) which is inspired by quantum annealing. However, this algorithm is designed for classical computers, while here we investigate the possible exploitation of a radically different computing machine, i.e., the D-Wave quantum computer, for biclustering.

The contributions of this paper can be summarized as follows: (1) We introduce the first QUBO model for the biclustering problem; more specifically, we formulate the biclustering problem as a repeated search for the most coherent biclusters following well-known approaches such as Cheng and Church (2000) and Ben-Dor et al. (2003), where biclusters are extracted one at a time from the data-matrix. (2) We analyze the model proving that it is correct, i.e., that the optimal solution of the QUBO model is the optimal solution for the one-bicluster problem. Results show that our model outperforms in terms of quality state-of-the-art biclustering approaches [i.e., BICRELS (Truong et al. 2013) and FLOC3 (Yang et al. 2005)]. (3) We discuss the practical applicability of our model by means of experiments performed on the D-Wave 2X™ architecture.

Overall, the key contribution of this work is a novel QUBO formulation for the biclustering problem that can be computed by quantum machines. Our investigation shows that such QUBO model is a viable approach for small-sized data matrices and the proposed principles might be used as a foundation for variant formulations better equipped to tackle larger datasets.

2 Background and related work

In this section, we first introduce the biclustering problem, then provide some necessary notions at the base of quantum annealing and the D-Wave architecture. Finally, we present the QUBO formalization for generic optimization problems.

2.1 Biclustering

As already mentioned, biclustering has been used in various application domains with different techniques. However, in its most general form, biclustering can be defined as the simultaneous clustering of rows and columns of a given data-matrix (Madeira and Oliveira 2004). The goal is then retrieving the subsets of rows and columns that have a coherent behavior, where “coherence” is defined according to the specific application domain (e.g., Euclidean distance, Pearson correlation).

In this paper, we formulate the problem of biclustering as a sequential search for the most coherent bicluster. This is a widely employed technique in the literature (Cheng and Church 2000; Ben-Dor et al. 2003; Denitto et al. 2017), and consists in extracting biclusters one by one from the data-matrix. Clearly, it is crucial how to “mask” the obtained bicluster before looking for the next one. There exist different heuristics in the literature addressing this problem: for example, one way to address this problem is to replace the obtained bicluster with background noise in the original data matrix (Cheng and Church 2000), so that the next bicluster can be looked for.

Hence, our problem takes as input a real-valued data matrix A with N rows and M columns, and returns a subset of rows and columns that identifies the most coherent bicluster. Each real value of the data-matrix $a_{i,j}$ encodes an "activation" level for a specific configuration. For example, for expression microarray data, rows typically represent genes and columns experimental conditions; hence, each entry $a_{i,j}$ represents the activation level of gene i under the experimental condition j. Our goal is to return the set of genes that exhibits a coherent behavior under the same subset of experimental conditions.

2.2 Quantum annealing and D-Wave

Among the various approaches to quantum information processing, a particularly interesting one is *adiabatic quantum optimization* and the closely related phenomenon of quantum annealing (QA), which allows us to replace exhaustive searches in global optimization problems with heuristic algorithms approximating the global optimum to the aim of finding a satisfactory solution. QA is a meta-heuristic based on the quantum adiabatic theorem,[1] whose basic strategy can be described as follows: first, the system is initialized to a simple state and then the conditions are slowly (adiabatically) changed to reach a complex final state that describes the solution to the computational problem of interest. The time-dependent Hamiltonian for QA is

$$H_{\text{QA}}(t) = A(t)H_{\text{init}} + B(t)H_{\text{prob}},$$

where the gradual transition from the ground state of the initial Hamiltonian H_{init} to the ground state of the problem Hamiltonian H_{prob} is defined by the annealing parameters $A(t)$ and $B(t)$ (Denchev et al. 2016). This is in some way similar to the classical simulated annealing (SA) (Farhi et al. 2002), which instead borrows a metaphor from the physical process used in metallurgy to create a defect-free crystalline solid. Rather then thermal fluctuations used in SA to control the search, in the quantum case, the computation is driven by *quantum fluctuations* and the tunneling field strength replaces temperature to control acceptance probabilities (Finnila et al. 1994). This is motivated by the fact that in SA the thermal transition probability depends only on the height of the potential wall to overcome, which means that in general it fails when it has to deal with very high barriers. The advantage of QA is the dependency of the tunneling probability both on the height and the width of the potential barrier, which gives it the ability to move in an energy landscape where local minima are separated by tall barriers, provided that they are narrow enough (Ray et al. 1989).

The QA optimization scheme has been implemented directly on quantum hardware by the Canadian company D-Wave Systems Inc. The D-Wave devices are able to minimize an objective function expressed in accordance to the Ising Model of statistical mechanics. This model can be arranged in a graph whose nodes are the spins and the edges represent interactions between them. The energy of the Ising model is expressed by the Hamiltonian

$$H(\sigma) = \sum_{\langle i\ j\rangle} J_{ij}\sigma_i\sigma_j + \sum_j h_j\sigma_j,$$

where $\sigma \in \{+1, -1\}$ and h_j is the external magnetic field in site j. The interaction between the spin in site i and the one in site j is given by J_{ij} and it can be either ferromagnetic ($J_{ij} < 0$, that tends to align spins) or anti-ferromagnetic ($J_{ij} > 0$, that tends to misalign spins). The Ising energy minimization problem is equivalent to the QUBO model presented in the next section. This means that solving the latter corresponds to finding the ground state energy of the associated Ising model (Bian et al. 2010).

Moreover, in order to solve an instance of a QUBO problem with a D-Wave machine, we need to adapt the logical formulation of a given problem (i.e., the logical Ising problem) to the physical fixed architecture of the quantum processor (i.e., the physical Ising problem). This architecture is composed by a matrix of unit cells (Fig. 1) that is a set of 8 qubits disposed in a bipartite graph. These unit cells are connected in a structure called *chimera graph*. At the time we are writing, the most recent version of the machine is the D-Wave 2000Q™which has 16 × 16 unit cells for a total of 2048 qubits.

2.3 Quadratic unconstrained binary optimization problems

The goal of a Quadratic Unconstrained Binary Optimization problem (QUBO) is to find the assignment to a set of binary

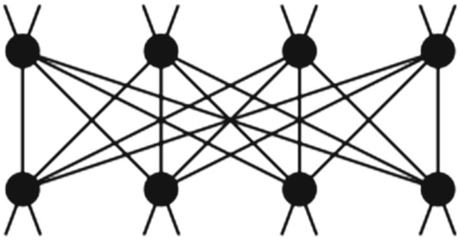

Fig. 1 D-Wave unit cell as shown in Dahl (2013)

[1] According to the quantum adiabatic theorem, a quantum system that begins in the non-degenerate ground state of a time-dependent Hamiltonian will remain in the instantaneous ground state provided the Hamiltonian changes sufficiently slowly.

variables $x_1 \ldots x_n$ so as to minimize an objective function of the form:

$$O(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i + \sum_{1 \leq i < j \leq n} b_{i,j} x_i x_j \tag{1}$$

We can also represent an instance of a QUBO problem with a weighted graph where each node represents a binary variable x_i, a linear coefficient a_i encodes the value associated with the node x_i and a quadratic coefficient $b_{i,j}$ represents the value associated with the edge between nodes x_i and x_j. With this representation, setting $x_i = 1$ corresponds to selecting the node x_i, while $x_i = 0$ corresponds to eliminating the node x_i from the graph. Hence, the objective function corresponds to the sum of all values in the graph and its minimization is equivalent to decide which nodes to remove (where removing a node implies the removal of all edges that are incident to that node), in such a way that the summation of the values remaining in the graph is the lowest possible.

3 The QUBO model for biclustering

In this section, we detail our QUBO model for the one-bicluster problem. We first describe a binary model for the one-bicluster problem; then, we show how such a model can be encoded as a QUBO.

3.1 A binary model for one-bicluster

We now present the objective function for the binary one-bicluster problem and in what follows we explain how it is derived. Given a real-valued data matrix A with N rows and M columns, the objective function for the binary one-bicluster problem is the following:

$$\underset{(c_{1,1},\ldots,c_{N,M})}{\arg\max} \left(\sum_{i,j} a_{i,j} c_{i,j} - \sum_{i,j,t,k} O_{i,j,t,k} c_{i,j} c_{t,k} + \sum_{i<t} B_{i,t} \right) \tag{2}$$

where $1 \leq i, t \leq N;\ 1 \leq j, k \leq M$.

In the first two terms, we have $N \times M$ binary variables $c_{i,j}$ that encode whether a given entry $a_{i,j}$ of the data matrix A belongs to the bicluster or not (where $c_{i,j} = 1$ indicates that the entry $a_{i,j}$ does belong to the bicluster).

Also, in this function, we can identify two forces: one that encourages points to group together, namely the first term in (2), and one that avoids points that are not coherent to be in the same group [i.e., the second term in (2)]. Such term is based on a value $O_{i,j,t,k}$ which measures the coherence between two points $a_{i,j}$ and $a_{t,k}$. The function $O_{i,j,t,k}$ depends on which kind of biclusters we wish to analyze. In particular, following the relevant literature (e.g., Tu et al. 2011), we consider two types of coherence:

Constant: Which aims at penalizing points that have a different activation level and hence identifies biclusters that have a single coherent value.

$$O_{i,j,t,k} = w|a_{i,j} - a_{t,k}| \tag{3}$$

Additive: Which identifies biclusters that encode an evolution of the activation values over columns.

$$O_{i,j,t,k} = w(a_{i,j} - a_{t,j} + a_{t,k} - a_{i,k})^2 \tag{4}$$

In both (3) and (4), the weight w can be adjusted to balance such two forces: setting w to high values favors the coherence of the points inside the biclusters, while setting w to low values favors the creation of large biclusters. The set of valid values for such weight is $\mathbb{R}^+$; however, setting high values could lead as a result to biclusters composed of a single element. The appropriate value to set depends on the data context and must be determined experimentally as shown in Denitto et al. (2017).

In order to solve our problem, we need to restrict the feasible variable assignments so that only valid assignments correspond to a bicluster. In other words, we need to rule out assignments that do not correspond to a subset of rows and columns that have all entries selected (see Fig. 2b for an example of a non-valid assignment). To do so, we add one constraint stating that, given two rows of the output matrix C, they have to share the same configurations or one of them must be zero. The constraint between rows i and t is expressed in Eq. (2) by the term:

$$B_{i,t} = \begin{cases} 0, & \text{if } (\sum_k c_{i,k} = 0) \vee (\sum_k c_{t,k} = 0) \\ & \quad \vee (\sum_k (c_{i,k} - c_{t,k}) = 0) \\ -\infty, & \text{otherwise} \end{cases} \tag{5}$$

Such constraint ensures that there is a permutation of rows and columns that forms a sub-matrix with all entries selected (i.e., visually a full rectangle of ones).

Another interesting way to look at an admissible configuration is that it can be described by fixing the same value for all the elements of a column with an exception for the elements that belong to a disabled row. For example, considering Fig. 2a (before permutations), the configuration can be expressed as: Columns $\{1, 3, 4\}$ take value 1, columns $\{2, 5\}$ take value 0 and row 2 is disabled (all the element are 0). Hence, any admissible configuration can be uniquely identified by this type of description. This description is useful to better understand the QUBO model we describe next.

$$C = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(a)

$$C = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

(b)

Fig. 2 Example of: a valid assignment and its permutation that results in a full rectangle of ones (**a**); an invalid assignment, no permutation can result in a full rectangle of ones (**b**)

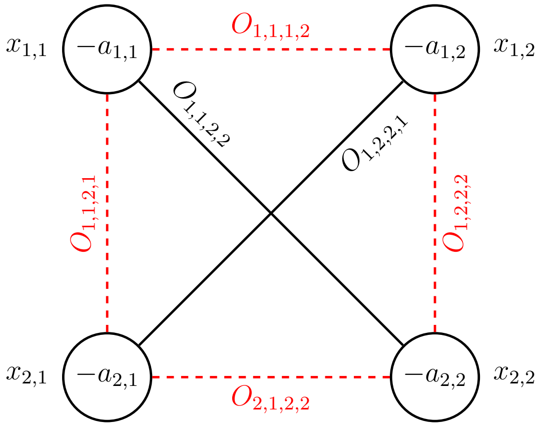

Fig. 3 A graphical representation of our QUBO model for a 2 × 2 data-matrix, the (red) dotted edges are absent in case of additive coherence measure (4) (color figure online)

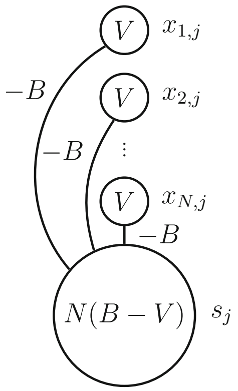

Fig. 4 Graphical representation of a column constraint

3.2 The QUBO model for the one-bicluster problem

We now provide a QUBO formulation for the binary model described above. For ease of explanation, let us start with a QUBO representation that does not consider the bicluster constraint [i.e., the $B_{i,t}$ elements in Eq. (2)]. To build such model by using the graph-based representation of QUBOs, we create a node $x_{i,j}$ for each variable $c_{i,j}$. Considering that the QUBO formulation has to be minimized, we then assign a coefficient $-a_{i,j}$ to each node. For each pair of nodes $(x_{i,j}, x_{t,k})$, we assign to the edge between them a positive value $O_{i,j,t,k}$ calculated according to the Eqs. (3) or (4). Note that the latter has value 0 for points on the same row or the same column, hence for such measure, the horizontal and vertical edges are absent from the graph. The corresponding objective function for the QUBO problem will then be:

$$\underset{(x_{1,1},\ldots,x_{N,M})}{\arg\min} \left(\sum_{i,j} -a_{i,j} x_{i,j} + \sum_{i,j,t,k} O_{i,j,t,k} x_{i,j} x_{t,k} \right) \tag{6}$$

where $1 \le i, t \le N;\ 1 \le j, k \le M$. It is easy to see that the assignment that maximizes function (2) without the bicluster constraint is the same that minimizes the QUBO objective function (6). Figure 3 shows a graphical representation of such a simplified QUBO model for a 2 × 2 input data matrix.

Now, in order to consider the bicluster constraint, we must add some extra nodes to the QUBO model so as to ensure that the assignments generated are valid (i.e., they represent a subset of rows and columns). As mentioned in Sect. 3.1, an admissible configuration should set all variables in the same column to the same value except for the variables that belong to disabled rows. To express this, we create two types of constraints: column constraints and row constraints. A column constraint ensures that all variables in a column have the same value (either 0 or 1). To do so, we add to each node a positive coefficient V and we add a new node to the graph with a coefficient equal to $N(B - V)$ where $B > V$. We call this new node the *column switch* and we indicate with s_j the variable that corresponds to the node switch for column j. Finally, we set the coefficient of the edges between the column switch and the N nodes to $-B$ (see Fig. 4 for a graphical representation). Intuitively, if k of the N nodes are selected and the switch is not active (i.e., $s_i = 0$), we add to the objective function a value kV. If we select the switch and the k nodes, we add $k(V - B) + N(B - V)$. Since we are minimizing the objective function the best configuration will be either selecting all nodes [with a contribution of $N(V - B) + N(B - V) = 0$] or not selecting any node (again with a contribution of zero). All other configurations will add a positive value to the objective function.

A row constraint should force all variables in a row to be zero when a specific condition holds (i.e., we decide to not consider that row). To enforce this, we add a new node to the graph with a coefficient 0 and we call this new node the *row switch*. We indicate with r_i the variable that corresponds to the node switch for row i (see Fig. 5 for a graphical representation). Then, we set the edges between the row switch and the M nodes to a positive coefficient G. Intuitively, when the $r_i = 0$ any configuration for the M nodes contributes with a null value to the objective function; hence, they are

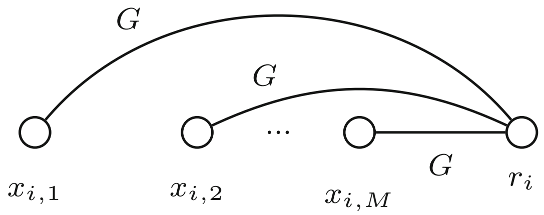

Fig. 5 Graphical representation of a row constraint

equally desirable. However, if $r_j = 1$, then selecting any of the M nodes will increase the objective function of a value G. Hence, in this case, the best configuration is the one that does not select any of the M nodes.

Finally we combine the first graph (Fig. 3) without the bicluster constraint (from now on called the *inner graph*) with the row and column constraints and by adding from each row switch to every column switch an edge with coefficient $V - B$. The objective function has now the following form:

$$\begin{aligned}\underset{(x_{1,1},\ldots,x_{N,M})}{\arg\min} \sum_{i,j} \Bigg(& V x_{i,j} - B x_{i,j} s_j + G x_{i,j} r_i \\ & + (V-B) r_i s_j + (B-V) s_j - a_{i,j} x_{i,j} \\ & + \sum_{t,k} O_{i,j,t,k} x_{i,j} x_{t,k} \Bigg)\end{aligned} \tag{7}$$

In order to ensure that our QUBO formulation is a proper model for the one-bicluster problem, we must show that for all valid solutions, the extra constraints (i.e., row and column constraints) contribute with a zero value, while for all non-valid solutions they contribute with a strictly positive value. In particular, we prove the following theorem:

Theorem 1 (Model validity) *Given a model of a data-matrix with N rows and M columns and values $B > V > 0$ and $G > B - V$, for all assignments that do not violate a row or a column constraint such extra constraints provide a null contribution to the objective function. For all other configurations, the contribution is > 0.*

Proof Given the objective function (7), we can observe that in each addend of the summation, the terms that depends from the combined constraint structure are:

$$V x_{i,j} - B x_{i,j} s_j + G x_{i,j} r_i + (V-B) r_i s_j + (B-V) s_j. \tag{7a}$$

Hence, each of these addend depend exclusively on three binary variables, namely a node from the inner graph $x_{i,j}$ and the two switches r_i and s_j. Now we compute the value of the term for the combined constraint structure equation (7a) exhaustively for all eight cases of the three variables:

1. $[x_{i,j} = 0, r_i = 0, s_j = 0]$: 0
2. $[x_{i,j} = 0, r_i = 0, s_j = 1]$: $B - V$
3. $[x_{i,j} = 0, r_i = 1, s_j = 0]$: 0
4. $[x_{i,j} = 0, r_i = 1, s_j = 1]$: $V - B + B - V = 0$
5. $[x_{i,j} = 1, r_i = 0, s_j = 0]$: V
6. $[x_{i,j} = 1, r_i = 0, s_j = 1]$: $V - B + B - V = 0$
7. $[x_{i,j} = 1, r_i = 1, s_j = 0]$: $V + G$
8. $[x_{i,j} = 1, r_i = 1, s_j = 1]$: $V - B + G + V - B + B - V = V - B + G$

For 1, 3, 4, 6 which represent a valid assignment where all the inner graph nodes are in compliance with the switches (i.e., do not violate row or a column constraints), the contribution is 0. For all the other configurations which represent a non-valid assignment, the contribution is greater that 0 (this is because $B > V > 0$ and $G > B - V$). □

In order to complete the model, we have to identify the appropriate values for V, B and G. To do so, we observe that a configuration that does not comply with all the switches constraints should increase more than the decrease in value that can derive from taking such a configuration in the inner graph, namely the values assigned to the structure should be high enough to ensure that the objective function does not minimize for the non-valid configurations. Although intuitively we can simply choose high values, to maintain the range of possible values as small as possible, we investigate what the lowest admissible ones are. Let us indicate with R a configuration for the row switches, S a configuration for the column switches, X a configuration for the inner graph nodes in compliance with the switches and $\overline{X}$ a configuration where any subset of X does not comply with the corresponding switches.

We can then show the following theorem:

Theorem 2 (Determining V, B, G) *Given the specific switches configurations R and S and the valid solution (X, R, S), we have that:*

$$\begin{gathered} O(\overline{X}, R, S) - O(X, R, S) > 0 \\ \Longleftrightarrow \\ \left(V > V_m \wedge B > B_m \wedge G > G_m\right) \end{gathered} \tag{8}$$

for all invalid solutions $(\overline{X}, R, S)$, where

$$\begin{aligned} V_m &= \max_{i,j}\{a_{i,j}\} \\ B_m &= V + \max_{i,j} \left\{ -a_{i,j} + \sum_{t,k} O_{i,j,t,k} \right\} \\ G_m &= B - V + \max_{i,j}\{a_{i,j}\} \end{aligned} \tag{9}$$

Proof Similarly to what we did for Theorem 1, we now compute the value of equation (7) for all configurations of the three binary variables $x_{i,j}$, r_i and s_j:

1. $[x_{i,j} = 0, r_i = 0, s_j = 0]$: 0
2. $[x_{i,j} = 0, r_i = 0, s_j = 1]$: $B - V$
3. $[x_{i,j} = 0, r_i = 1, s_j = 0]$: 0
4. $[x_{i,j} = 0, r_i = 1, s_j = 1]$: $V - B + B - V = 0$
5. $[x_{i,j} = 1, r_i = 0, s_j = 0]$: $V - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k}$
6. $[x_{i,j} = 1, r_i = 0, s_j = 1]$: $V - B + B - V - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k} = -a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k}$
7. $[x_{i,j} = 1, r_i = 1, s_j = 0]$: $V + G - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k}$
8. $[x_{i,j} = 1, r_i = 1, s_j = 1]$: $V - B + G + V - B + B - V - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k} = V - B + G - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k}$

In order to ensure the desired behavior, the difference between a non eligible configuration $(\overline{X}, R, S)$ and an eligible configuration (X, R, S) must be higher than 0. Let us impose this condition to the difference between the previous eight cases:

- **[5]–[1]** $> 0 \Rightarrow V - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k} > 0$
- **[7]–[3]** $> 0 \Rightarrow V + G - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k} > 0$
- **[8]–[4]** $> 0 \Rightarrow V - B + G - a_{i,j} + \sum_{t,k} O_{i,j,t,k} x_{t,k} > 0$
- **[2]–[6]** $> 0 \Rightarrow B - V + a_{i,j} - \sum_{t,k} O_{i,j,t,k} x_{t,k} > 0$

Because the coherence measure $O_{i,j,t,k}$ is always greater or equal to 0, we are now ready to determine the minimum value to assign to V, B and G.
From the first difference [5]–[1], we have:

$$V > \max_{i,j}\{a_{i,j}\} = V_m$$

From the last difference [2]–[6] we have that:

$$B > V + \max_{i,j}\left\{-a_{i,j} + \sum_{t,k} O_{i,j,t,k}\right\} = B_m$$

And from the third one [8]–[4] we have:

$$G > B - V + \max_{i,j}\{a_{i,j}\} = G_m$$

The second one [7]–[3] holds because of V and G already defined. □

3.3 Properties of the model

Theorems 1 and 2 ensure that, by building the model as described above, for any valid configuration (i.e., a configuration that describes a bicluster), the contribution of the column and row constraints to the objective function is null. For all valid assignments the objective function reported in (7) reduces to (6), hence the configuration that minimizes (7) is the same that maximizes equation (2) (i.e., the most coherent bicluster). Moreover, for any non-valid assignment (i.e., an assignment that does not encode a bicluster) the contribution of the row and column constraints will be strictly positive hence such configuration will always be discarded in favor of a valid assignment.

The proposed model can be further simplified. In particular, we can reduce the number of edges (quadratic terms) by observing that if a couple of nodes (in the inner graph) on different rows and columns are active (i.e., two nodes on the opposite corners of a rectangle) also the other two nodes on the other diagonal of the rectangle must be active to comply with the switches. The terms $O_{i,j,t,k} x_{i,j} x_{t,k}$ and $O_{t,j,i,k} x_{t,j} x_{i,k}$ either contribute both or none to the objective function. Hence, we can add both values $O_{i,j,t,k} + O_{t,j,i,k}$ to a single edge and remove the other one. Hence, regardless of the coherence measure used, we can remove half of the diagonal edges. An example of the complete simplified model is shown in Fig. 6.

As for space complexity, given an input matrix $N \times M$, the model has $NM + N + M$ binary variables. The number of edges depends on the coherence metric used. In particular, for the constant coherence Eq. (3), we have in the worst case (i.e., when all the coherence measures are different from 0) $NM(NM-1)/2 - NM(N-1)(M-1)/4 + 3NM$ edges. For the additive coherence Eq. (4), we must insert into the model only the diagonal edges (see Fig. 6); hence, the total number of edges, in the worst case, is $NM(N-1)(M-1)/4 + 3NM$.

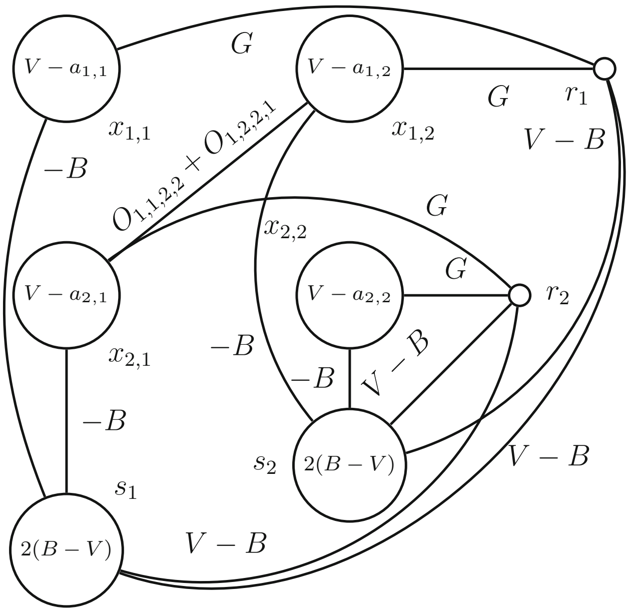

Fig. 6 Graph of the complete model for $N = 2$ and $M = 2$ with the additive coherence similarity metric (4) and the simplification proposed at the end of this section

Since the complexity class of the problem is NP-complete (hence the problem is in general not tractable), our main motivation for the work is to investigate the possibility to exploit the quantum annealing process to combat such a complexity. Based on the above analysis, the main computational bottleneck for our model is space requirements. While the worst-case analysis reveals a polynomial complexity for what concerns space, typical application domains for biclustering can involve data matrix with a large number of rows and columns (i.e., thousands of genes and hundreds of experiments). For such numbers, the space requirement for our model becomes an issue that needs an adequate treatment. To this purpose, in the next section, we also propose a sparsification method in order to simplify the model by eliminating a given percentage of edges using a heuristic. Moreover, following previous approaches (Denitto et al. 2014), we use a decomposition technique in order to aggregate biclusters extracted from sub-matrices.

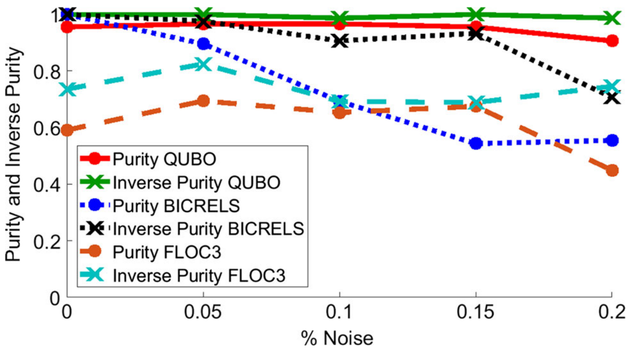

Fig. 7 Performance comparison of our QUBO model, BICRELS and FLOC3 varying noise level

4 Empirical evaluation of the model

Having described and analyzed our approach, we now present an empirical evaluation of our QUBO model for biclustering. In what follows we first describe the methodology we use to perform the experiments and then we present results obtained by following established evaluation protocols for biclustering (Tu et al. 2011).

4.1 Evaluation methodology

The main goals of our empirical evaluation are: (1) validate the accuracy of the QUBO model for biclustering comparing it with state-of-the-art approaches [BICRELS (Truong et al. 2013) and FLOC3 (Yang et al. 2005)]; (2) evaluate how the removal of edges from the model affects the quality of the solutions; (3) evaluate the quality of our model through a widely exploited biclustering dataset (Prelić et al. 2006); (4) assess the applicability of the model on current state-of-the-art quantum processing units (i.e., the D-Wave architecture).

Hence, we created a synthetic dataset so to accurately measure the performance of our approach. In particular, the dataset is composed by 10×10 matrices with a constant random-positioned bicluster that occupies the 25 percent of the elements. Then, we added a Gaussian noise to each matrix, where the standard deviation of such Gaussian noise is a percentage of the difference between the mean of the entries belonging to the biclusters and the mean of all the others. In particular, we considered 5 different percentage values from 0 (no noise) to 0.2. We generated a set of 15 matrices per noise level for a total of 75 matrices. This dataset allows us to measure the accuracy of the algorithms by comparing the bicluster extracted from the models with the ground-truth (i.e., the bicluster that is present in the data-matrix). Given C the set of elements of the bicluster found and L the set of elements of the real bicluster, to measure such accuracy we use two established metrics (Tu et al. 2011):

- $Purity = |C \cap L|/|C|$ which represents how many elements of the solution belong to the real bicluster.
- $InversePurity = |C \cap L|/|L|$ which represents how many elements of the real bicluster have been found.

4.2 Validating the accuracy of the QUBO model

For each of the 75 matrices of the dataset, the QUBO form has been solved by using the CPLEX library (V12.6) and by applying 24 different weights w (constant for each $O_{i,j,t,k}$) to the similarity measure (4), for a total of 1800 tests.

Here, we present the results of Purity and Inverse Purity as a function of the noise level. For each noise level, we analyzed and set the parameters of the procedures with the values that gives the best average result on the 15 matrices with that noise level. Please note that the optimal value of w, which influences the size of the ideal biclusters, depends on the data context and has to be determined empirically. Solving each instance takes milliseconds; hence, the overhead to determine the optimal value of w is not an issue. Note that this is the same protocol used in Denitto et al. (2017). Results in Fig. 7 show that our QUBO model significantly outperforms BICRELS and FLOC3 in terms of quality of the bicluster extracted.

4.3 Sparsification of the model

We can observe that the model exhibits some degree of redundancy. In particular not all the edges in the inner graph, that is all the similarity measurements between points of the input matrix, affect with the same weight the selection of the optimal solution. For example, assume we know the sub-matrix that forms the most coherent bicluster, intuitively many of the edges internal to such sub-matrix will have a low value (because the elements of the bicluster are coherent); hence,

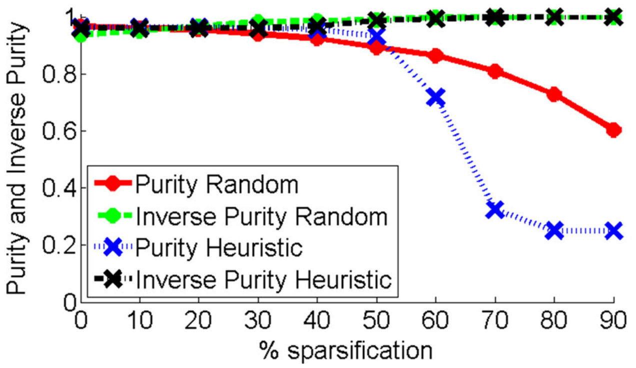

Fig. 8 Average of purity and inverse purity varying noise level for both random and heuristic sparsifications described in Sect. 4.3

most of such edges could be removed from the model. For this reason, we tested the removal of different percentages of edges from the inner graph of the 1800 instances previously described, namely we removed from 0 to 90% (with steps of 10%) of the edges for a total of 18,000 tests.

Note that sparsification is only intended to be a practical heuristic to address larger matrix, but we cannot provide any guarantees on how this affect optimality. In contrast, our aim is to investigate whether simple sparsification heuristics could maintain a good level of accuracy while providing significant reductions in space for the model. While assess which edges are redundant (without knowing the bicluster) is not straightforward, the empirical evaluation shows that some simple heuristics do provide a significant gain. In more detail, our procedure for sparsification computes a value for each internal edge and then sort edges according to such value. We then remove the first $X\%$ (where $X \in \{0, 10, 20 \ldots 90\}$) of these edges. We tried different values for the edges that are all based on a combination of the function $O_{i,j,t,k}$ and the values of the matrix entries that relates to this function (i.e., $a_{i,j}$, $a_{t,k}$, $a_{i,k}$, $a_{t,j}$). Moreover, we compare such heuristics with a random approach where we remove $X\%$ (where $X \in \{0, 10, 20 \ldots 90\}$) of the total edges choosing randomly between all internal edges in the model. Figure 8 reports a comparison of the best heuristic with the random approach. The values computed by this heuristic is the ratio: $\frac{O_{i,j,t,k}}{a_{i,j}+a_{t,k}+a_{i,k}+a_{t,j}}$. Overall, our results confirm that with a simple heuristic, one can achieve similar level of accuracy with approximately half the edges of the QUBO model.

4.4 Evaluation on benchmarking data-set (Prelić et al. 2006)

We evaluated our model on the benchmarking synthetic dataset introduced in Prelić et al. (2006).[2] The matrices proposed in that dataset contains 100×50 entries. Such matrices cannot be directly analyzed by our approach due to the space complexity associated with our model (see Sect. 3.3). However, following previous approaches (Denitto et al. 2014), we can extract biclusters from sub-matrices and then aggregate the results. In particular, in our experiments, we consider a 10×10 window that selects a portion of the data matrix and we shift this windows over the data with a full coverage and an overlap degree of 5 rows/columns. We call each sub-matrix a *kernel*. The proposed protocol consists of the following three steps:

1. *Generate the bicluster set* We extract one bicluster from each kernel using the additive coherence.
2. *Aggregate the results* We group the biclusters provided by step 1 by using a similarity based clustering algorithm [Affinity Propagation (Frey and Dueck 2007)]. We defined as similarity between two biclusters the number of rows/columns they share.
3. *Retrieve the final bicluster* Please notice that the coherency in biclusters obtained at the previous step is not guaranteed. For this reason, we assign to each bicluster a score, exploiting the objective function (Eq. 2), i.e., evaluating the objective function for such bicluster. This step is repeated for all groups obtained in step 2 and by keeping the best solution (according to the objective function) we keep the most coherent solution.

The accuracy of the resulting biclusters has been assessed with the same metrics used in Prelić et al. (2006) (i.e., the *Gene Match Score*). Results in Fig. 9 shows that our method is competitive with other state-of-the-art approaches [see Fig. 2a in Prelić et al. (2006)], confirming the potentials of the proposed approach.

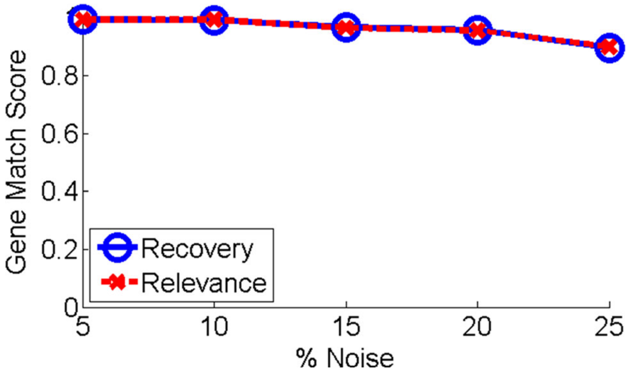

Fig. 9 Performance evaluation on benchmarking dataset as explained in Sect. 4.4

[2] Available at http://www.tik.ee.ethz.ch/sop/bimax (Scenario I—Noise).

5 D-Wave experiments

In this section, we report results of experiments performed on a D-Wave 2X™ machine. The D-Wave 2X™ machine that we used is hosted at NASA Ames Research Laboratory and has 12 × 12 unit cells for a total of 1152 qubits,[3] see Denchev et al. (2016) for more details on its hardware and performance.

First, in Sect. 5.1, we describe the embedding of the problem into the D-Wave 2X™ hardware. In Sect. 5.2, we make some considerations about the criteria for the tractability of the biclustering problem with this quantum annealer. Then, in Sect. 5.3, we discuss the results of the embedding phase, and finally in Sect. 5.4, we describe the outcomes obtained with this machine.

5.1 Embedding the QUBO model on the D-Wave architecture

As previously mentioned in Sect. 2.2 in order to solve a QUBO model on a D-Wave machine, we need to adapt the formulation of a problem to the physical fixed architecture of the quantum processor. Different problems require different connectivity and in order to embed a problem into the architecture we can either formulate the QUBO model taking into account the fixed structure of the hardware graph, or create a logical formulation (as we did in our QUBO model) and then embed the logical graph into the physical one through the minor embedding technique.

The minor embedding process determines a mapping of the physical qubits into the problem's variables, i.e., which physical qubits should represent which variable of the logical QUBO formulation. Heuristics in order to determine this mapping has been developed and more details on the minor embedding techniques for the D-Wave can be found in Cai et al. (2014). Note that, even if the number of nodes of the model is smaller than the number of qubits of the processor, it is not always possible to find a valid embedding. In particular, the embedding into the hardware architecture usually requires more variables, since some nodes are represented by several physical qubits (a "chain" of qubits) due to the sparse connectivity of the hardware graph. All the experiments described here have been performed by applying the embedding process to our model using the official D-Wave libraries.

The parameters we used in the embedding are those standard provided by the D-Wave. Moreover, we perform only a single embedding attempt with standard parameters. This approach, also followed in other papers, e.g., O'Gorman et al. (2015b), is based on the Cai heuristics mentioned before which may be very suboptimal. As done in Venturelli et al. (2015), Pudenz (2016) and Perdomo-Ortiz et al. (2015), we could study an optimal choice of the parameters that is more appropriate for the biclustering problem. This may lead to a better performance of the D-Wave on our problem.

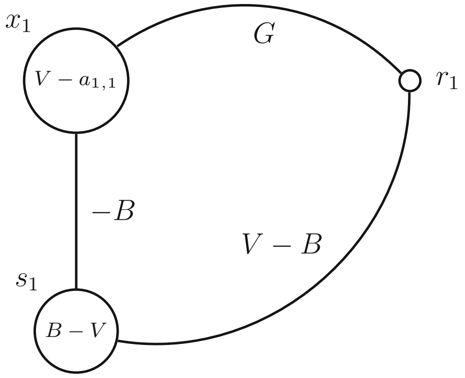

Fig. 10 Graph of the complete model for $N = 1$ and $M = 1$

5.2 Suitability of D-Wave for biclustering problem

Before tackling an optimization problem with a quantum annealing device, it is crucial to ensure that the problem shows (King et al. 2017):

- *global frustration*,[4] i.e., it requires a non trivial combinatorial optimization,
- *local ruggedness*, i.e., the problem presents a landscape with tall and thin barriers.

As biclustering is known to be NP-complete, we expect that its logical Ising/QUBO formulation straightforwardly displays global frustration. In fact, it is possible to show such a behavior even in the limit of a 1-dimensional matrix biclustering, which can be seen as the building block of any biclustering instance, cf. Fig. 10.

In this trivial case, the geometry of the problem is reduced to a complete graph with three vertices. A frustrated behavior, with two ferromagnetic couplings and an anti-ferromagnetic one, prevails when the magnitudes of the weights associated with the three edges become of the same order,[5] i.e., when the B parameter is significantly larger than V.

If we consider an arbitrary $N \times M$ biclustering instance, frustration increases because of the presence of $N \cdot M$ of the triangular loops of Fig. 10 which share vertices among themselves. Moreover, as we can see from Eq. (9), since B has a linear dependence on N and M, while V does not, this

[3] Note that only 1097 of 1152 qubits are operational.

[4] An Ising model is frustrated when the competition between ferromagnetic and anti-ferromagnetic couplings leads to a ground state where the interaction energies between spins cannot be simultaneously minimized.

[5] Note that parameter G can always be chosen close to B.

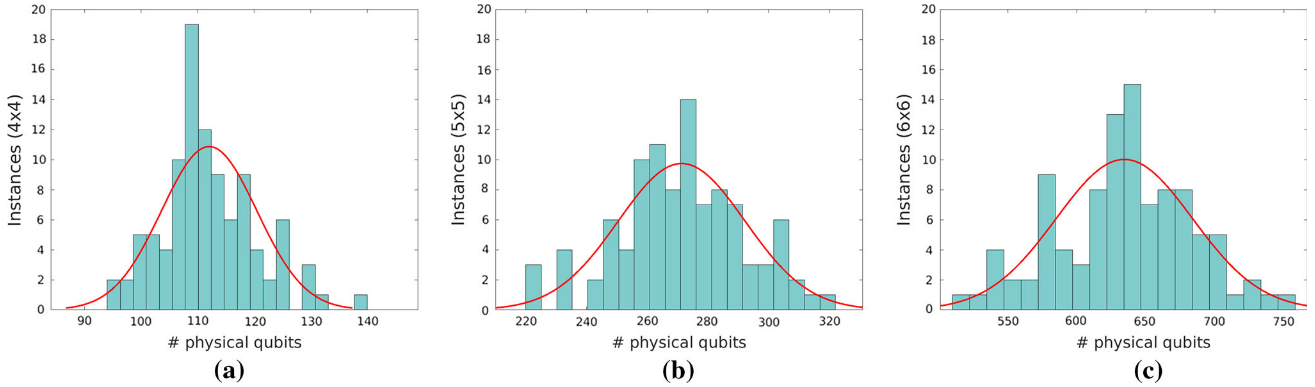

Fig. 11 Results of the embedding phase: number of physical qubits required to embed **a** the 4 × 4 instances, **b** the 5 × 5 instances and **c** the 6 × 6 instances

automatically pushes the model in a $V \ll B$ highly frustrated regime when increasing N or M.

Usually, the complex landscape typical of frustrated systems only guarantees the presence of many local minima and maxima and it does not imply that barriers separating them are tall and narrow enough for QA to work properly. In our biclustering model, such condition of local ruggedness is ensured by the QUBO formulation, since the geometry of the problem guarantees the formation of clusters of nodes which are internally ferro-magnetic coupled (King et al. 2017). This feature, from the point of view of the energy landscape, translates into the presence of high and narrow barriers separating minima. To summarize, the complexity of our biclustering model ensures a macroscopically interesting landscape with multiple local minima (global frustration) and the particular geometry of the problem guarantees the high and narrow barriers in the landscape (local ruggedness).

5.3 Embedding phases

For the real experiments, we randomly generated the following instances (matrices) for the biclustering problem:

- 100 instances of a size of 4 × 4 and with bicluster of 2 × 2
- 100 instances of a size of 5 × 5, 50 of which with a bicluster 2 × 3 and 50 with a bicluster 3 × 2
- 100 instances of a size of 6 × 6 and with bicluster of 3 × 3

All these instances are without noise and from these we generated the QUBO models using the additive coherence measure (Eq. 4) with a weight parameter $w = 1$. Results of the number of physical qubits required after this embedding phase can be observed in the histograms with a Gaussian distribution fit in Fig. 11a for the 4 × 4 instances, Fig. 11b for the 5 × 5 instances and Fig. 11c for the 6 × 6 instances. Also in Table 1, we report the aggregated results with mean and standard deviation.

Table 1 Results of the embedding phase: number of physical qubits required to embed an instance

Size	Min	Max	μ	σ
4 × 4	94	139	112.03	8.45
5 × 5	220	321	271.33	20.90
6 × 6	511	757	634.37	49.41

We can observe that the number of physical qubits required grows significantly as the instance size increases. With just a starting matrix of 6 × 6, we already require almost half of the available physical qubits. As previously mentioned, this is due to the fact that the few available connection between physical qubits on the D-Wave architecture necessarily lead to the use of a high number of physical qubits to represent a single logical qubits. In fact, our biclustering model consists of fully connected sub-components which lead to a quadratic overhead even for the most efficient embedding (Boothby et al. 2016). More details on the number of physical qubits required to represent a single logical one after the embedding phase can be observed in Fig. 12a for the 4 × 4 instances, Fig. 12b for the 5 × 5 instances and Fig. 12c for the 6 × 6 instances. Also in Table 2 we report the aggregated results with mean and standard deviation. As reported in all the cases, for some logic qubits, the embed requires a minimum of 1 physical qubits; however, the maximum number required grows as the instance dimension increases. Specifically, the maximum number required for some qubits in the 6 × 6 instances is 30, which it is three times the maximum required number for the 4 × 4 instance.

5.4 D-Wave experiments results

The objective of this experimental phase is to determine whether the D-Wave 2X™ machine is able to retrieve the optimum solution of the QUBO objective functions of the

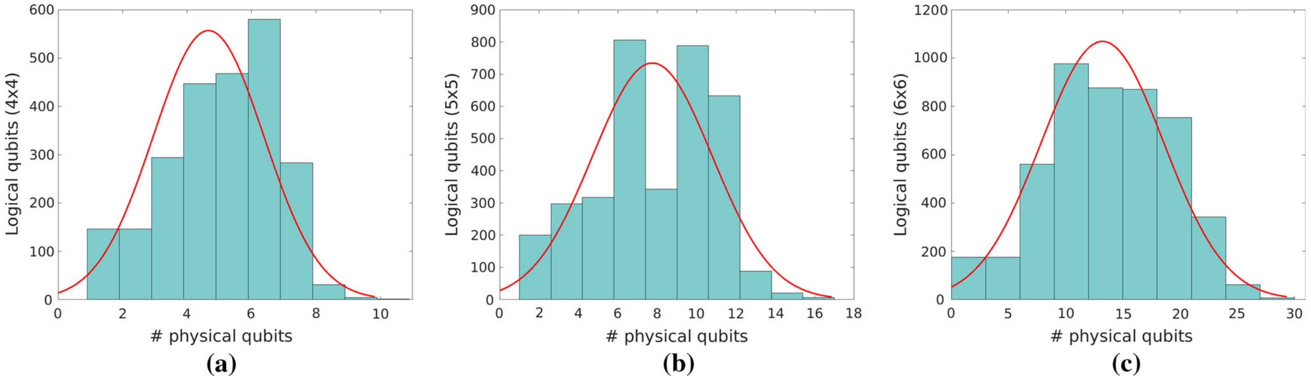

Fig. 12 Results of the embedding phase: number of physical qubits per logical qubit in **a** the 4 × 4 instances, **b** the 5 × 5 instances and **c** the 6 × 6 instances

Table 2 Results of the embedding phase: number of physical qubits per logical qubit varying instance size

Size	Min	Max	μ	σ
4 × 4	1	10	4.67	1.72
5 × 5	1	17	7.75	3.04
6 × 6	1	30	13.22	5.37

instances previously described. The D-Wave takes as input the number of reads a *num_reads* parameter which identifies the number of states (output solutions) to read from the solver in each programming cycle (which we set as described later) along with other hardware specific parameter which we kept as the default values of the machine (e.g., the default annealing time for every read of 20 microseconds). In this experimental phase, we solved every instance previously described with the following protocol:

- We solved the QUBO instance using the CPLEX library in order to find the configuration that gives the optimum of the objective function.
- We run the instance on the D-Wave 2X™ machine. Specifically, we run a programming cycle asking for 10,000 reads. Hence, the D-Wave samples the objective function 10,000 times and returns the 10,000 solutions.
- We process the sampled solutions comparing them to the one obtained with CPLEX, in order to check if the optimum has been found.
- If the optimum has not been found, we repeat the process with a new programming cycle (we set the maximum iteration to 1000 cycle; however, it was not necessary to perform so many cycles as can be seen in the following results).

Regarding the 4 × 4 instances, as we can observe in Fig. 13, we obtained most of optimum solution in just one programming cycle and no more than 4 cycles was required to solve all the 100 instances. As expected, the number of cycles required grows as the instance size increases. In more details, regarding the 5 × 5 matrices, we also obtained most of the times the optimum in one cycle and solved all 100 instances in no more than 55 cycles and regarding the 6 × 6 we always obtained the optimum solution in less then 550 cycles. Specifically, just one 6 × 6 instance required 550 runs, we solved all the other 99 instances in up to 228 cycles. We also report in Fig. 14 the average number of cycles required per instance size. These results lead to the conclusion that it was always possible to get the optimal solution for all generated QUBO instances.

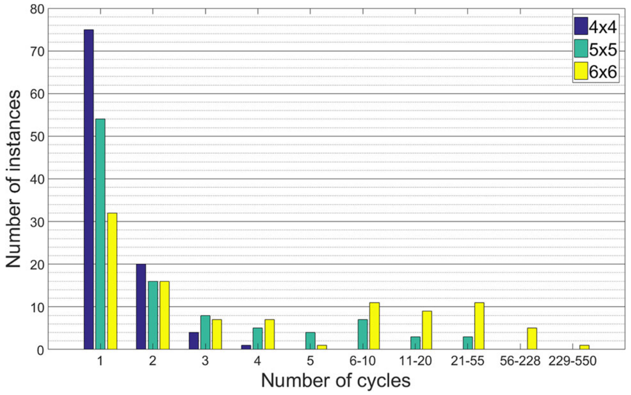

Fig. 13 Histogram of the number of instances where the optimum of the objective function has been found after a specific number of programming cycles, varying the instance size

As previously done in Rieffel et al. (2015), we compute the probability of success Ps for a 20 µs annealing time (which is the annealing time we used for a single read). For each set of instances of the same dimension, we then compute the expected number of runs $k = \frac{\ln(0.01)}{\ln(1-Ps)}$ required to obtain a 99% success probability and multiply it for 20 µs to compute the total annealing time required to obtain a 99% success. Results are shown in Fig. 15.

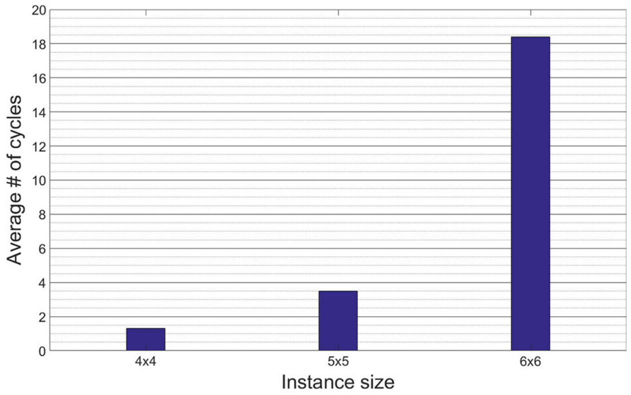

Fig. 14 Average number of programming cycles required to find the optimum solution of the QUBO objective function varying the instance size

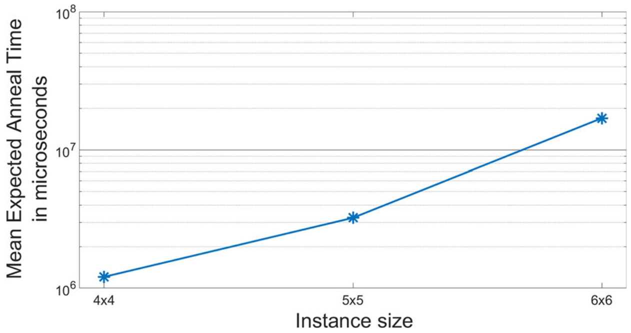

Fig. 15 Expected total annealing time in microseconds to 99% success for the three instance sizes

6 Conclusions and future works

In this paper we investigated, the possible use of quantum annealing for solving biclustering problems. In particular, we introduced a novel QUBO model for the one-bicluster problem and show its correctness. As for the practical applicability of quantum annealing to biclustering, we have tested our model by means of real experiments on a D-Wave 2X™ machine. Results suggest that the use of a quantum annealing approach is feasible only for small matrices. This is due to the current D-Wave architecture. We believe that further developments of the D-Wave machine including the use of a larger number of qubits with higher connectivity could allow us to practically use quantum annealing for hard real-world problems involving biclustering. Thus, this paper takes a first important step toward the effective use of quantum annealing for solving the biclustering problem.

Our future works includes: (1) the investigation of different formulations of the QUBO model that do not require auxiliary variables in order to embed larger instances; (2) the use of sophisticated representations techniques to ameliorate the limitations imposed by current experimental hardware (Bian et al. 2014). Moreover, we will investigate the use of frameworks to test the statistical significance of the discovered biclusters by filtering the solutions with state-of-the-art statistical tests (Henriques and Madeira 2018).

Acknowledgements We acknowledge the support of the Universities Space Research Association (USRA) Quantum Artificial Intelligence Laboratory Research Opportunity program. We would like to thank in particular Davide Venturelli for his very helpful comments on a first draft of this paper.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Ayadi W, Elloumi M, Hao J (2012) BiMine+: an efficient algorithm for discovering relevant biclusters of DNA microarray data. Knowl Based Syst 35:224–234

Badea L (2009) Generalized clustergrams for overlapping biclusters. In: Proceedings of the 21st international joint conference on artificial intelligence. IJCAI'09. Morgan Kaufmann Publishers Inc., San Francisco, pp 1383–1388

Ben-Dor A, Chor B, Karp R, Yakhini Z (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. J Comput Biol 10(3–4):373–384

Bian Z, Chudak F, Macready WG, Rose G (2010) The Ising model: teaching an old problem new tricks. http://www.dwavesys.com/sites/default/files/weightedmaxsat_v2.pdf

Bian Z, Chudak F, Israel R, Lackey B, Macready WG, Roy A (2014) Discrete optimization using quantum annealing on sparse Ising models. Front Phys 2:56

Bicego M, Lovato P, Ferrarini A, Delledonne M (2010) Biclustering of expression microarray data with topic models. In: International conference on pattern recognition (ICPR2010), pp 2728–2731

Boothby T, King AD, Roy A (2016) Fast clique minor generation in chimera qubit connectivity graphs. Quantum Inf Process 15(1):495–508

Cai J, Macready WG, Roy A (2014) A practical heuristic for finding graph minors. ArXiv e-prints arXiv:1406.2741

Cheng Y, Church G (2000) Biclustering of expression data. In: Proceedings eighth international conference on intelligent systems for molecular biology (ISMB00), pp 93–103

Dahl ED (2013) Programming with D-Wave: map coloring problem. http://www.dwavesys.com/sites/default/files/MapColoringWP2.pdf

Denchev VS, Boixo S, Isakov SV, Ding N, Babbush R, Smelyanskiy V, Martinis J, Neven H (2016) What is the computational value of finite-range tunneling? Phys Rev X 6(3):031015

Denitto M, Farinelli A, Franco G, Bicego M (2014) A binary factor graph model for biclustering. In: Frnti P, Brown G, Loog M, Escolano F, Pelillo M (eds) Structural, syntactic, and statistical pattern recognition, vol 8621. Lecture notes in computer science. Springer, Berlin, pp 394–403

Denitto M, Farinelli A, Figueiredo MA, Bicego M (2017) A biclustering approach based on factor graphs and the max-sum algorithm. Pattern Recognit 62:114–124

Dhillon I (2001) Coclustering documents and words using bipartite spectral graph partitioning. In: Proceedings of international conference on knowledge discovery and data mining, pp 269–274

Dolnicar S, Kaiser S, Lazarevski K, Leisch F (2012) Biclustering: overcoming data dimensionality problems in market segmentation. J Travel Res 51(1, (1)):41–49

Farhi E, Goldstone J, Gutmann S (2002) Quantum adiabatic evolution algorithms versus simulated annealing. Eprint arXiv:quant-ph/0201031

Finnila AB, Gomez MA, Sebenik C, Stenson C, Doll JD (1994) Quantum annealing: a new method for minimizing multidimensional functions. Chem Phys Lett 219:343–348

Flores JL, Inza I, Larranaga P, Calvo B (2013) A new measure for gene expression biclustering based on non-parametric correlation. Comput Methods Programs Biomed 112(3):367–397

Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976

Henriques R, Madeira SC (2014) BicPAM: pattern-based biclustering for biomedical data analysis. Algorithms Mol Biol 9(1):27

Henriques R, Madeira SC (2018) BSig: evaluating the statistical significance of biclustering solutions. Data Min Knowl Disc 32(1):124–161. https://doi.org/10.1007/s10618-017-0521-2

Henriques R, Antunes C, Madeira SC (2015) A structured view on pattern mining-based biclustering. Pattern Recognit 48(12):3941–3958

Kadowaki T, Nishimori H (1998) Quantum annealing in the transverse Ising model. Phys Rev E 58(5):5355–5363

King J, Yarkoni S, Raymond J, Ozfidan I, King AD, Nevisi MM, Hilton JP, McGeoch CC (2017) Quantum annealing amid local ruggedness and global frustration. ArXiv e-prints arXiv:1701.04579

Kochenberger G, Hao J, Glover F, Lewis M, Lü Z, Wang H, Wang Y (2014) The unconstrained binary quadratic programming problem: a survey. J Comb Optim 28(1):58–81

Kurihara K, Tanaka S, Miyashita S (2009) Quantum annealing for clustering. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. UAI '09. AUAI Press, Arlington, pp 321–328

Madeira S, Oliveira A (2004) Biclustering algorithms for biological data analysis: a survey. IEEE Trans Comput Biol Bioinform 1:24–44

Mukhopadhyay A, Maulik U, Bandyopadhyay S, Coello C (2014) Survey of multiobjective evolutionary algorithms for data mining: part II. IEEE Trans Evolut Comput 18(1):20–35

Neven H, Rose G, Macready WG (2008) Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization, ArXiv e-prints arXiv:0804.4457

Oghabian A, Kilpinen S, Hautaniemi S, Czeizler E (2014) Biclustering methods: biological relevance and application in gene expression analysis. PLoS ONE 9(3):e90,801

O'Gorman B, Babbush R, Perdomo-Ortiz A, Aspuru-Guzik A, Smelyanskiy V (2015a) Bayesian network structure learning using quantum annealing. Eur Phys J Spec Top 224(1):163–188

O'Gorman B, Rieffel E, Do M, Venturelli D, Frank J (2015b) Compiling planning into quantum optimization problems: a comparative study. In: Proceedings of the workshop on constraint satisfaction techniques for planning and scheduling problems (COPLAS-15), pp 11–20

Perdomo-Ortiz A, Fluegemann J, Biswas R, Smelyanskiy VN (2015) A performance estimator for quantum annealers: Gauge selection and parameter setting. ArXiv e-prints arXiv:1503.01083

Prelić A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9):1122–1129

Pudenz KL (2016) Parameter setting for quantum annealers. In: 2016 IEEE high performance extreme computing conference (HPEC), pp 1–6

Ray P, Chakrabarti BK, Chakrabarti A (1989) Sherrington–Kirkpatrick model in a transverse field: absence of replica symmetry breaking due to quantum fluctuations. Phys Rev B 39:11,828–11,832

Rieffel EG, Venturelli D, O'Gorman B, Do MB, Prystay EM, Smelyanskiy VN (2015) A case study in programming a quantum annealer for hard operational planning problems. Quantum Inf Process 14:1–36 arXiv:1407.2887

Santoro GE, Tosatti E (2006) Optimization using quantum mechanics: quantum annealing through adiabatic evolution. J Phys A Math Gen 39(36):R393–R431

Truong DT, Battiti R, Brunato M (2013) A repeated local search algorithm for biclustering of gene expression data. In: Hancock E, Pelillo M (eds) Similarity-based pattern recognition. Springer, Heidelberg, pp 281–296. https://doi.org/10.1007/978-3-642-39140-8_19

Tu K, Ouyang X, Han D, Honavar V (2011) Exemplar-based robust coherent biclustering. In: SDM, SIAM, pp 884–895

Venturelli D, Mandrà S, Knysh S, O'Gorman B, Biswas R, Smelyanskiy V (2015) Quantum optimization of fully connected spin glasses. Phys Rev X 5(031):040

Yang J, Wang H, Wang W, Yu PS (2005) An improved biclustering method for analyzing gene expression profiles. Int J Artif Intell Tools 14(05):771–789