A Quantum Annealing Approach to Biclustering

Lorenzo Bottarelli, Manuele Bicego, Matteo Denitto, Alessandra Di Pierro(✉), and Alessandro Farinelli

University of Verona, Strada le Grazie 15, 37134 Verona, Italy
alessandra.dipierro@univr.it

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^{TM} computer, whose processor is able to exploit quantum mechanical effects in order to perform quantum annealing. The question motivating this work is whether the use of this special hardware is a viable approach to efficiently solve the biclustering problem. As a first step towards the solution of this problem, we show a feasible encoding of biclustering into the D-Wave^{TM} quantum annealing hardware, and provide a theoretical analysis of its correctness.

1 Introduction

Biclustering, also known in other scenarios as co-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 [19]. It is used in several different scenarios, such as document analysis [11], market segmentation [12], recommender systems [20] and, most importantly, expression microarray data analysis [2,15,19,22,24]. 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. Addressing this issue can provide invaluable information to biologists, given the ever increasing amount of data that they have to analyse.

Different biclustering techniques have been proposed in the past [1,5,7,10,27], each one characterized by different features, such as computational complexity, effectiveness, interpretability and optimization criterion – cf [15,19,24]) for a general review. A significant part of these approaches aim at adapting a given clustering technique to the biclustering problem, for example by repeatedly performing rows and columns clustering. However, an interesting recent chunk of works aim at proposing novel models for biclustering, where rows and columns

C. Martín-Vide et al. (Eds.): TPNC 2016, LNCS 10071, pp. 175–187, 2016.
DOI: 10.1007/978-3-319-49001-4_14

are analysed simultaneously (as opposed to clustering rows and columns separately) [27]. 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 biclustering instance is represented by a matrix with thousands of column/rows [19]. Moreover, the underlying optimization task required by the model is inherently intractable 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 metaheuristic, called Quantum Annealing (QA) [14,16,26], to the global optimization problems underlying biclustering, by following some recent developments in the construction of quantum devices that physically realise quantum annealing. Similarly to the classical Simulated Annealing, QA is an optimization metaheuristic that seeks the global optimum of an objective function by following a process inspired by the thermodynamic process of annealing. In this search, QA is more effective than the classical method as it 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 [9]. 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 [21], Bayesian network structure learning [23] and hard operational planning problems [25]. As done in [25] or in [21] for image recognition, we show here an encoding of biclustering as a Quadratic Unconstrained Binary Optimization (QUBO) problem [17], 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^{TM} 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 [18] 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^{TM} 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 largest biclusters following well known approaches such as [4,7], where biclusters are extracted one at a time from the data-matrix. (2) We analyse 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. (3) We discuss the practical applicability of our model considering the current architecture of the D-Wave^{TM} machine.

2 Background and Related Work

In this section we first detail the biclustering problem, then provide some necessary notions on quantum annealing and on the D-wave^{TM} 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 [19]. 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, etc.).

In this paper we follow a standard technique [4,7], where biclusters are extracted one by one from the data-matrix. In particular, we focus on the problem of finding the largest bicluster in a data matrix, which in the rest of the paper we will refer to as the *one-bicluster* problem. The other biclusters can then be retrieved by masking the bicluster(s) already found (i.e. setting to a predefined value the relevant entries of the data-matrix) and by iterating the process.

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 largest, 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^{TM}

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

[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.

and then the conditions are slowly (adiabatically) changed to reach a complex final state that describes the solution to a computational problem of interest. It is in some way similar to the classical simulated annealing (SA) [13], 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 [14]. The QA optimization scheme has been implemented directly on quantum hardware by the Canadian company D-Wave Systems Inc. The D-Wave$^{\text{TM}}$ devices are able to minimize an objective function expressed in accordance to the Ising Model of statistical mechanics. The Ising energy minimization problem is NP-hard [3] and it is equivalent to the QUBO model presented in the next section.

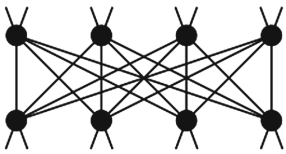

Fig. 1. D-Wave$^{\text{TM}}$ unit cell as shown in [8]

In order to solve an instance of a QUBO problem with a D-Wave$^{\text{TM}}$ machine we will need to adapt the logical formulation of a given problem to the physical fixed architecture of the quantum processor. 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*.

The current version of the machine (D-Wave 2X$^{\text{TM}}$) has 12×12 unit cells for a total of 1152 qubits (cf [9] for more details on its hardware and performance).

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 variables $x_1 ... x_n$ so as to minimize an objective function that has the following form:

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

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 to the node x_i and a quadratic coefficient $b_{i,j}$ represents the value associated to 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:

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

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 analyse. In particular, following the relevant literature (e.g., [27]) 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.

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

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

(a) (b)

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

In both (3) and (4) the weight w can be adjusted to balance such two forces: setting w to high values favours the coherence of the points inside the biclusters while setting w to low values favours the creation of large biclusters. Such weight must be determined experimentally and it is domain dependent.

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:

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

Such constraint ensures that there is a permutation of rows and columns that forms a submatrix 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.

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 value $-a_{i,j}$ 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:

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

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

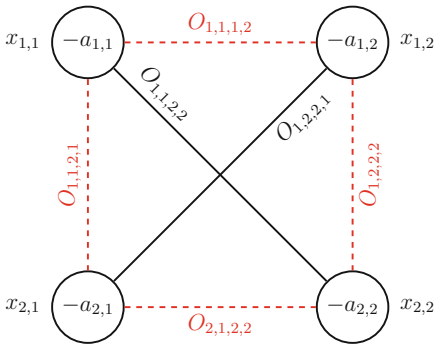

Fig. 3. A graphical representation of our QUBO model for a 2×2 data-matrix, the (red) dotted edges are absent in case of additive coherence measure (4).(Color figure online)

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. To do so we add to each node a positive value V and we add a new node to the graph with a value 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 value of the edges between the column switch and the N nodes to $-B$ (see Fig. 4a 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 minimising

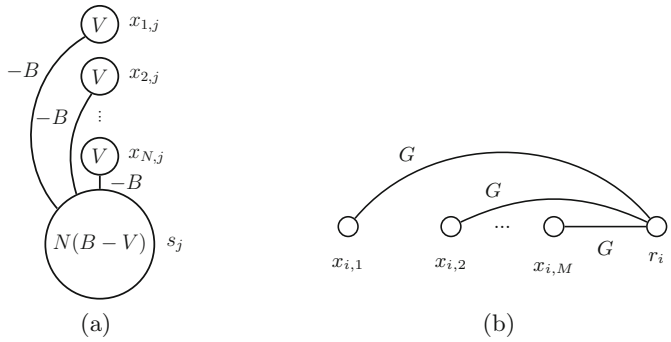

Fig. 4. Graphical representation of: a column constraint (4a); a row constraint (4b)

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 value 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. 4a for a graphical representation). Then, we set the edges between the row switch and the M nodes to a positive value 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 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 value $V-B$. The objective function has now the following form:

$$\arg \min_{(x_{1,1},\cdots,x_{N,M})} \sum_{i,j} \Big(Vx_{i,j} - Bx_{i,j}s_j + Gx_{i,j}r_i + (V-B)r_i s_j + (B-V)s_j - a_{i,j}x_{i,j} + \sum_{t,k} O_{i,j,t,k}x_{i,j}x_{t,k} \Big) \tag{7}$$

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*

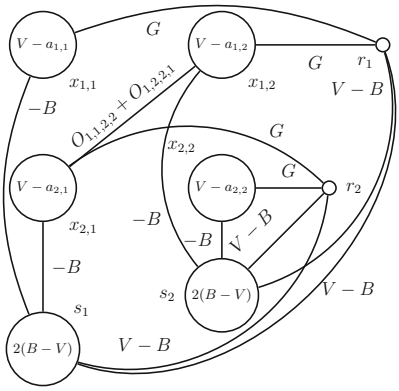

Fig. 5. Graph of the complete model for $N = 2$ and $M = 2$ with the additive coherence similarity metric (4) and the simplification proposed at the end of this section.

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 greater than 0.

Proof (Proof sketch). From Eq. (7) we can observe that in each addend of the summation, the terms that depend on the constraint structure are $Vx_{i,j} - Bx_{i,j}s_j + Gx_{i,j}r_i + (V - B)r_is_j + (B - V)s_j$. Hence, each addend depends exclusively on three binary variables, namely a node from the inner graph $x_{i,j}$ and the two switches r_i and s_j. By analyzing exhaustively the eight cases we can reach the desired conclusion.

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:*

$$O(\overline{X}, R, S) - O(X, R, S) > 0 \iff \big(V > V_m \wedge B > B_m \wedge G > G_m\big) \quad (8)$$

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

$$V_m = \max_{i,j}\{a_{i,j}\} \qquad B_m = V + \max_{i,j}\{-a_{i,j} + \sum_{t,k} O_{i,j,t,k}\}$$
$$G_m = B - V + \max_{i,j}\{a_{i,j}\} \tag{9}$$

Proof (Proof sketch). In a similar way to what we explained in Theorem 1, we can analyze the cases of Eq. (7) considering the configurations of the three binary variables $x_{i,j}$, r_i and s_j. We can then calculate the values V_m,B_m,G_m by imposing $O(\overline{X}, R, S) - O(X, R, S) > 0$ between the eight cases.

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 Eq. (7) reduces to Eq. (6), hence the configuration that minimizes Eq. (7) is the same that maximizes Eq. (2) (i.e., the most coherent, largest 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. 5.

3.3 Minor Embedding the QUBO Model on the D-Wave™ Architecture

In order to solve a QUBO model on a D-Wave™ machine, an arbitrary logical graph has to be embedded into the physical structure of the processor. This requires a mapping of the physical qubits into the problem's variables, i.e. to determine which physical qubits should represent which variable of the QUBO problem. In order to perform this operation we adopt the approach developed in [6] for finding graph minors. More specifically, we use a minor embedding technique. 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 due to the sparse connectivity of the hardware graph.

We applied this minor embedding technique to our model using the code provided by authors of [6] with the aim to determine the dimension of the matrices and the sparsification required to always find a valid embedding. Results show that the dimensions of the matrices that can be analysed on D-Wave^{TM} are significantly smaller than typical data for biclustering. Specifically, we can embed matrices up to 7×7 on the processor of the current version of the D-Wave^{TM} machine (1152 qubits). Nonetheless, it is possible to decompose large matrices into smaller ones, achieving good results in terms of accuracy for the retrieved biclusters.

4 Conclusions

In this paper we have introduced a QUBO model for the one-bicluster problem. The results suggest that nowadays the use of such an approach would be possible only for small matrices on the current processor of the D-Wave^{TM} machine as explained in Sect. 3.3. Nonetheless, this paper takes a first important step towards an effective use of quantum annealing for biclustering.

Several challenges will need to be addressed for the development of future improved quantum annealers. Critical issues such as longer coherence times, calibration accuracy, etc. are objects of intense study by researchers in both academy and industry as well as by the quantum enhanced optimization (QEO) group of the Intelligence Advanced Research Projects Agency (IARPA) of the US government. Solving these issues will hopefully lead in a near future to the implementation of quantum annealing that will be more flexible both in terms of connectivity and choice of cost function, thus adding value to studies like the one proposed in this paper.

References

1. Ayadi, W., Elloumi, M., Hao, J.: Bimine+: An efficient algorithm for discovering relevant biclusters of DNA microarray data. Knowl. Based Syst. **35**, 224–234 (2012)
2. Badea, L.: Generalized clustergrams for overlapping biclusters. In: IJCAI, pp. 1383–1388 (2009)
3. Barahona, F.: On the computational complexity of Ising spin glass models. J. Phys. A: Math. Gen. **15**(10), 3241–3253 (1982)
4. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: The order-preserving submatrix problem. J. Comput. Biol. **10**(3–4), 373–384 (2003)
5. Bicego, M., Lovato, P., Ferrarini, A., Delledonne, M.: Biclustering of expression microarray data with topic models. In: International Conference on Pattern Recognition (ICPR2010), pp. 2728–2731 (2010)
6. Cai, J., Macready, W.G., Roy, A.: A practical heuristic for finding graph minors. ArXiv e-prints, June 2014
7. Cheng, Y., Church, G.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB00), pp. 93–103 (2000)

8. Dahl, E.D.: Programming with D-Wave: Map Coloring Problem (2013). http://www.dwavesys.com/sites/default/files/Map%20Coloring%20WP2.pdf
9. Denchev, V.S., Boixo, S., Isakov, S.V., Ding, N., Babbush, R., Smelyanskiy, V., Martinis, J., Neven, H.: What is the computational value of finite-range tunneling? Phys. Rev. X **6**(3), 031015 (2016)
10. Denitto, M., Farinelli, A., Franco, G., Bicego, M.: A binary factor graph model for biclustering. In: Fränti, P., Brown, G., Loog, M., Escolano, F., Pelillo, M. (eds.) S+SSPR 2014. LNCS, vol. 8621, pp. 394–403. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44415-3_40
11. Dhillon, I.: Coclustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 269–274 (2001)
12. Dolnicar, S., Kaiser, S., Lazarevski, K., Leisch, F.: Biclustering : overcoming data dimensionality problems in market segmentation. J. Travel Res. Q. Publ. Travel Tourism Res. Assoc. **51**(1), 41–49 (2012)
13. Farhi, E., Goldstone, J., Gutmann, S.: Quantum Adiabatic Evolution Algorithms versus Simulated Annealing. eprint, January 2002. arXiv:quant-ph/0201031
14. Finnila, A.B., Gomez, M.A., Sebenik, C., Stenson, C., Doll, J.D.: Quantum annealing: A new method for minimizing multidimensional functions. Chem. Phys. Lett. **219**, 343–348 (1994)
15. Flores, J.L., Inza, I., Larranaga, P., Calvo, B.: A new measure for gene expression biclustering based on non-parametric correlation. Comput. Methods Programs Biomed. **112**(3), 367–397 (2013)
16. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. Phys. Rev. E **58**(5), 5355–5363 (1998)
17. Kochenberger, G., Hao, J., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. J. Comb. Optim. **28**(1), 58–81 (2014)
18. Kurihara, K., Tanaka, S., Miyashita, S.: Quantum annealing for clustering. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 321–328. UAI 2009, AUAI Press, Arlington, Virginia, United States (2009)
19. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: a survey. IEEE Trans. Comput. Biol. Bioinform. **1**, 24–44 (2004)
20. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Coello, C.: Survey of multiobjective evolutionary algorithms for data mining: Part ii. Evol. Comput. IEEE Trans. **18**(1), 20–35 (2014)
21. Neven, H., Rose, G., Macready, W.G.: Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization, ArXiv e-prints, April 2008
22. Oghabian, A., Kilpinen, S., Hautaniemi, S., Czeizler, E.: Biclustering methods: Biological relevance and application in gene expression analysis. PLoS ONE **9**(3), e90801 (2014). http://dx.doi.org/10.1371%2Fjournal.pone.0090801
23. O'Gorman, B., Babbush, R., Perdomo-Ortiz, A., Aspuru-Guzik, A., Smelyanskiy, V.: Bayesian network structure learning using quantum annealing. Eur. Phys. J. Spec. Top. **224**(1), 163–188 (2015). http://dx.doi.org/10.1140/epjst/e2015-02349-9
24. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9), 1122–1129 (2006)

25. Rieffel, E.G., Venturelli, D., O'Gorman, B., Do, M.B., Prystay, E.M., Smelyanskiy, V.N.: A case study in programming a quantum annealer for hard operational planning problems. Quantum Inform. Process. **14**, 1–36 (2015)
26. Santoro, G.E., Tosatti, E.: Optimization using quantum mechanics: quantum annealing through adiabatic evolution. J. Phys. A Math. Gen. **39**(36), R393–R431 (2006)
27. Tu, K., Ouyang, X., Han, D., Honavar, V.: Exemplar-based robust coherent biclustering. In: SDM, pp. 884–895. SIAM (2011)