

OPEN

# A QUBO Formulation of Minimum Multicut Problem Instances in Trees for D-Wave Quantum Annealers

William Cruz-Santos<sup>1</sup>, Salvador E. Venegas-Andraca<sup>2\*</sup> & Marco Lanzagorta<sup>3</sup>

Quantum annealing algorithms were introduced to solve combinatorial optimization problems by taking advantage of quantum fluctuations to escape local minima in complex energy landscapes typical of NP – hard problems. In this work, we propose using quantum annealing for the theory of cuts, a field of paramount importance in theoretical computer science. We have proposed a method to formulate the Minimum Multicut Problem into the QUBO representation, and the technical difficulties faced when embedding and submitting a problem to the quantum annealer processor. We show two constructions of the quadratic unconstrained binary optimization functions for the Minimum Multicut Problem and we review several tradeoffs between the two mappings and provide numerical scaling analysis results from several classical approaches. Furthermore, we discuss some of the expected challenges and tradeoffs in the implementation of our mapping in the current generation of D-Wave machines.

Quantum Annealing algorithms (QA) constitute a paradigm of quantum computation focused on solving combinatorial optimization problems<sup>1–9</sup>. QA are based on quantum effects in order to escape local minima of a cost function by the effect of tunneling through barriers separating local minima. An example of physical realizations of quantum annealing are the D-Wave quantum computers which have proved to be advantageous to solve instances of some NP – hard problems<sup>10</sup>.

To solve a problem using the D-Wave architecture, we must express it as a quadratic unconstrained Boolean optimization problem (QUBO) or an equivalent Ising function defined on logical variables. Then, we embed the logical problem in the physical architecture of the quantum annealer by mapping logical variables and qubits. The final step consists of performing an annealing process and obtaining the results. There are some limitations of the D-Wave hardware such as the maximum number of physical qubits available (which is 2000 qubits in its most recent computer, the D-Wave 2000Q system); this limitation imposes a restriction on the size of the logical problem that can be embedded into the hardware. Another limitation is the connectivity of the current architecture which is in the form of a Chimera graph as, in order to represent a logical variable, it is necessary to use a chain of physical qubits. Finally, numerical precision is another key feature in the mathematical definition and computation of Ising model parameters due to the analog nature of the device and the presence of additive noise.

In this article, we present a QUBO transformation of a family of graphs, to be later used on a QA approach, in order to solve the Minimum Multicut Problem (MCC problem) which is of paramount importance in theoretical computer science and other disciplines such as computer vision. The MMC problem is defined as follows: given a graph  $G = (V, E)$  with associated positive weights to the edges and a list of vertex pairs  $(s_i, t_i)$ ,  $1 \leq i \leq k$ , a minimum weight set of edges separating each pair of vertices in the list must be found.

The MMC problem is an NP – hard problem. In this paper, we study a QUBO formulation of the MCC problem on a family of connected trees, being this variant also NP – hard. We show two QUBO constructions for the studied problem. The first one is based on an intuitive linear programming formulation used to obtain a function of degree equal to the largest unique path between each pair  $(s_i, t_i)$ ,  $1 \leq i \leq k$  in a tree. The second one is based on the basic idea that in order to separate each path between terminals  $s_i$  and  $t_i$ , it is necessary to remove at most  $k$  edges. This basic observation was used to define a QUBO function of degree at most  $k - 1$ . Also, a general penalty approach for the QUBO function based on Boolean circuits is shown.

<sup>1</sup>CU-UAEM Valle de Chalco, Hermenegildo Galeana 3, Valle de Chalco, Estado de México, 56615, Mexico. <sup>2</sup>Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Ave. Eugenio Garza Sada 2501, Monterrey, NL, 64849, Mexico. <sup>3</sup>US Naval Research Laboratory, 4555 Overlook Ave. SW, Washington, DC, 20375, USA. \*email: [svenegas@tec.mx](mailto:svenegas@tec.mx)

### Results

**Quantum annealing, pseudo boolean functions and QUBO formulations.** The D-Wave hardware is a physical realization of QA which solves instances of the classical Ising problem on a transverse field. The Ising model problem is NP – hard<sup>11</sup> and it is defined as follows: given a set of weights (called *fields*)  $h_i$  and  $J_{ij}$  (called *couplers*), find an assignment to the set of Ising spin variables  $\mathbf{s} = \{s_i; 1 \leq i \leq N\}$ , with  $s_i \in \{-1, +1\}$ , so as to minimize the energy function

$$E(\mathbf{s}) = \sum_{1 \leq i \leq N} h_i s_i + \sum_{1 \leq i < j \leq N} J_{ij} s_i s_j, \tag{1}$$

$|h_i| \leq 2$  and  $|J_{ij}| \leq 1$ . Finding an assignment  $\mathbf{s}^* = \arg \min_{\mathbf{s} \in \{-1, +1\}^N} E(\mathbf{s})$  is equivalent to finding the ground state of the corresponding Ising classical Hamiltonian,

$$H_p = \sum_{1 \leq i \leq N} h_i \sigma_i^z + \sum_{1 \leq i < j \leq N} J_{ij} \sigma_i^z \sigma_j^z \tag{2}$$

where  $\sigma_i^z$  is the Pauli matrix  $z$  acting on the  $i$ th particle. One of the schemes to realize QA is through adiabatic quantum evolution from the ground state of an initial Hamiltonian to a ground state of a final Hamiltonian<sup>5</sup>. According to this scheme a time-dependent Hamiltonian takes the form

$$H(\tau) = A(\tau)H_0 + B(\tau)H_p$$

where  $\tau = t/t_a$  for  $0 \leq t \leq t_a$ , is  $t_a$  is the total annealing time, and the initial Hamiltonian  $H_0 = -\sum_{1 \leq i \leq N} \sigma_i^x$  responsible for quantum tunneling among the localized classical states corresponding to the eigenstates of Hamiltonian  $H_p$ . Functions  $A(\tau)$  and  $B(\tau)$  are defined so that, at time  $\tau = 0$ , the influence of Hamiltonian  $H_0$  is predominant against  $H_p$ . As time evolution goes from  $\tau = 0$  to  $\tau = 1$ , the influence of Hamiltonian  $H_p$  increases while  $H_0$  fades away.

Consider the Hamiltonian  $H(t) = \tilde{H}(t/t_a) = \tilde{H}(\tau)$  such that  $0 \leq \tau \leq 1$ , and let us denote by  $|l; \tau\rangle$  the instantaneous eigenvector of  $\tilde{H}(\tau)$  corresponding to the instantaneous eigenvalue  $\lambda_l(\tau)$ . We find that

$$\tilde{H}(\tau)|l; \tau\rangle = \lambda_l(\tau)|l; \tau\rangle$$

with

$$\lambda_0(\tau) \leq \lambda_1(\tau) \leq \dots \leq \lambda_{2^N-1}(\tau).$$

The Adiabatic theorem asserts that for sufficiently large  $t_a$ ,

$$\lim_{t_a \rightarrow \infty} |\langle l = 0; \tau = 1 | \psi(t_a) \rangle| = 1 \tag{3}$$

for some solution  $\psi(t)$  to the Schrödinger equation with Hamiltonian  $H(t)$ . Consequently, state  $\psi(t_a)$  will be very close to the ground state of Hamiltonian  $H_p$  with high probability. A sufficient condition for the algorithm running time needed to satisfy the Adiabatic theorem is

$$t_a \gg \frac{\varepsilon_{\max}}{g_{\min}^2}$$

where

$$g_{\min} = \min_{0 \leq \tau \leq 1} (\lambda_1(\tau) - \lambda_0(\tau)) \quad \text{such that} \quad \lambda_1(1) \neq \lambda_0(1)$$

and

$$\varepsilon_{\max} = \max_{0 \leq \tau \leq 1} \left| \langle l = 1; \tau | \frac{d\tilde{H}}{d\tau} | l = 0; \tau \rangle \right|$$

*Pseudo-boolean maps.* Let  $\mathbf{x} = \{x_i; 1 \leq i \leq N\}$  be a set of  $N$  Boolean variables. A *pseudo-Boolean function* is a map  $f: \{0, 1\}^N \rightarrow \mathbb{R}$  represented as multi-linear polynomials:

$$f(\mathbf{x}) = \sum_{S \subseteq \{1, \dots, N\}} c(S) \prod_{j \in S} x_j, \tag{4}$$

where  $c(S) \in \mathbb{R}$ .

The *degree* of  $f$ , denoted as  $\text{deg}(f)$ , is the cardinality of the largest subset  $S \subseteq \{1, \dots, N\}$  for which  $c(S) \neq 0$ . Similarly, the *size* of  $f$  denoted as  $\text{size}(f)$  is the total number of variable occurrences in it, i.e.,  $\text{size}(f) = \sum_{S: c(S) \neq 0} |S|$ .

Of our particular interest are the *quadratic* pseudo-Boolean functions  $f_{ue}: \{0, 1\}^N \rightarrow \mathbb{R}$  with  $\text{deg}(f_{ue}) \leq 2$  expressed by polynomials of the form:

$$f_{ue}(\mathbf{x}) = \sum_{1 \leq i \leq N} u_i x_i + \sum_{1 \leq i < j \leq N} e_{ij} x_i x_j \tag{5}$$

where  $u_i, e_{ij} \in \mathbb{R}$ . The quadratic unconstrained binary optimization problem (QUBO) thus consists in finding an assignment  $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \{0,1\}^N} f_{ue}(\mathbf{x})$ . Also, the quadratic pseudo-Boolean map coincides with the Ising model via variable substitution  $x_i = (1 + s_i)/2$  for  $i = 1, \dots, N$ .

It is often the case that a problem can be formulated in terms of pseudo Boolean expressions of degree greater than two, which can subsequently be reduced to a QUBO function. However, this reduction commonly implies adding new variables to the former problem<sup>12–14</sup> (an introduction to NP-hard problems and quantum annealing-based algorithms can be found in<sup>15</sup>).

In the following sections, we address the QUBO formulation of the MMC problem which is of utmost theoretical and practical importance in the computer science community.

**Quantum formulation of the MMC problem.** An undirected weighted graph has the form  $G = (V, E, w)$  where  $V$  is a finite set of vertices,  $E \subseteq V^{(2)} = \{\{u, v\} | u, v \in V\}$  is a set of unordered pairs of vertices or edges and  $w: E \rightarrow \mathbb{R}^+ \cup \{0\}$  is a weighted map. Let us now define the MCC problem.

**Problem 1 (MMC problem)** Given a weighted graph  $G = (V, E, w)$  and a list of vertices pairs  $(s_i, t_i)$ ,  $1 \leq i \leq k$ , find a multicut with minimum weight, i.e., a subset  $E' \subseteq E$  such that the removal of  $E'$  from  $E$  disconnects  $s_i$  from  $t_i$  for every pair  $(s_i, t_i)$ , where the weight of  $E'$  is given as  $\sum_{\{u, v\} \in E'} w(u, v)$ .

For  $k = 1$ , the MMC problem reduces to the min-cut/max-flow problem that can be solved in polynomial time<sup>16</sup>. The problem is also tractable when  $k = 2$ , by using two applications of the min-cut/max-flow algorithm<sup>17</sup>. It becomes NP-hard when  $k \geq 3$  for general graphs, but can be solved in polynomial time for planar graphs for any fixed  $k$ <sup>18</sup>. It has been proved that the MMC problem is NP-hard and MAXSNP-hard on trees<sup>18,19</sup>. The MAXSNP-hardness of the MCC problem implies that no polynomial time approximation scheme exists unless  $P = NP$ .

The MMC problem can be applied in many areas such as telecommunication, routing, VLSI design and circuit partitioning<sup>20,21</sup>. In the following, we study the QUBO formulation of the MMC problem for the case when  $G$  is a tree.

*A direct mapping of MMC into QUBO.* Let  $G = (V, E, w)$  be a weighted graph and  $(s_i, t_i)$ ,  $1 \leq i \leq k$  a list of vertex pairs. For each edge  $e \in E$ , we introduce a Boolean variable  $x_e$  such that  $x_e = 0(1)$ , if  $e$  is (not) considered for a MMC in  $G$ . If graph  $G$  is restricted to be a tree, then there exists a unique path in  $G$  for every pair of vertices. We define the unique path from  $s_i$  to  $t_i$  in  $G$  as  $p_i$  where its length  $l_i$  is equal to the number of edges that it crosses. The diameter of a tree  $T = (V, E)$ , denoted as  $diam(T)$ , is equal to the maximum path length between every pair of vertices in  $T$ .

Let  $P = \bigcup_{i=1}^k p_i$  be the union of all edges in the paths  $p_i$  for  $i = 1, \dots, k$ . Let us to define the following function,

$$H = H_w + H_{pen} \quad (6)$$

where

$$H_w = \sum_{e \in P} w(e)(1 - x_e) \quad (7)$$

and

$$H_{pen} = \lambda_p \sum_{i=1}^k \prod_{e \in p_i} x_e. \quad (8)$$

In (6),  $H_w$  expresses the weight of the selected edges to be removed and  $H_{pen}$  serves to add a penalty value when the considered edges do not correspond to a MMC. Based on this construction, the MMC problem on trees is equivalent to minimizing  $H$  over all possible assignments to the Boolean variables  $x_e$ .

The penalty term satisfies  $0 \leq H_{pen} \leq k\lambda_p$ , where  $\lambda_p$  is a positive constant. In particular, when all paths are disconnected,  $H_{pen} = 0$  which means that at least one edge was removed in every path  $p_i$  for  $i = 1, \dots, k$ . We must ensure that  $\lambda_p$  is big enough so that the weight of an invalid multicut is greater than the weight of any valid multicut. The value of  $\lambda_p$  can be upper bounded by  $\lambda_p = \sum_{e \in P} w(e)$ . The degree of  $H_{pen}$  is equal to the maximum length of the paths  $p_i$  for  $i = 1, \dots, k$ . In other words,

$$\deg(H_{pen}) = \max \{l_i | i = 1, \dots, k\}.$$

In order to optimize (6) via QA, we need to write it in QUBO form (possibly by using section 1.1). For instance, assume that each path  $p_i$  does not have a trivial length, i.e.  $l_i > 2$ ; using the reduction method in<sup>14</sup>, the penalty term  $H_{pen}$  can be transformed into a quadratic function adding a total of  $\sum_{i=1}^k \left\lfloor \frac{l_i - 1}{2} \right\rfloor$  new variables. This transformation is as follows:

- If  $l_i$  is even then

$$\prod_{e \in p_i} x_e = S_2 + \min_{y \in \{0,1\}^{k_1}} \{B - 2AS_1\} \quad (9)$$

- If  $l_i$  is odd then we find that

$$\prod_{e \in p_i} x_e = S_2 + \min_{y \in \{0,1\}^{k_1}} \{B - 2AS_1 + y_{k_1}(S_1 - l_i + 1)\} \tag{10}$$

where

$$S_1 = \sum_{e \in p_i} x_e, \quad S_2 = \sum_{e, e' \in p_i, e \neq e'} x_e x_{e'}, \quad A = \sum_{j=1}^{k_1} y_j, \quad B = \sum_{j=1}^{k_1} (4j - 1)y_j, \tag{11}$$

and  $\mathbf{y} = (y_1, \dots, y_{k_1}) \in \{0, 1\}^{k_1}$  is a vector of  $k_1$  new variables, and  $k_1 = \lfloor \frac{l_i - 1}{2} \rfloor$ .

The proposed QUBO function obtained from (6) using the Ishikawa reduction method presented in<sup>14</sup> has as main advantage that its coefficients are small. This is a highly desirable property when programming the D-Wave quantum annealer because of its limited hardware precision to specify the values of Ising model parameters  $h_i$  and  $J_{ij}$ .

A disadvantage of function  $H$  given in (6) is that its degree depends on the length of the paths. In subsection 2.2 we present another method to build a low degree  $H_{pen}$ , based on the number of edges shared between paths.

*Construction of  $H_{pen}$  based on crossing paths.* Although several techniques exist for degree reduction of an arbitrary pseudo-Boolean function into a quadratic one, it is preferable to construct an initial expression of degree as low as possible. In this method, the key idea behind building a low degree function  $H_{pen}$  is to notice that a multicut consists of a subset of edges of cardinality of at most  $k$ . This condition imposes a restriction on the number of edges that could be removed in order to disconnect each path. Based on this observation, our goal will be to construct a new penalty function  $H'_{pen}$  such that it will penalize all subsets of edges of cardinality larger than  $k$ .

For any  $i, j \in \{1, \dots, k\}$ , let  $\zeta_{ij}$  be a characteristic function defined as

$$\zeta_{ij} = \begin{cases} 1 & \text{if } p_i \cap p_j \neq \emptyset, \quad i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We say that two paths  $p_i$  and  $p_j$  intersect if  $\zeta_{ij} = 1$ . In particular, if  $i = j$  then  $\zeta_{ij} = 0$ . For each  $j \in \{1, \dots, k\}$ , the quantity  $c_j = \sum_{i=1}^k \zeta_{ij}$  is equal to the number of paths  $p_i$  that intersect with path  $p_j$ . Thus, the maximum number of paths than can intersect with a path  $p_j$  is  $k - 1$ . It is assumed that the number of intersections cannot be greater than the length of a given path.

For any  $j \in \{1, \dots, k\}$  and  $\eta \in \{1, \dots, k - 1\}$ , let  $C_{\eta j}$  be given by

$$C_{\eta j} = \left( \eta - l_j + \sum_{e \in p_j} x_e \right)^2 \tag{13}$$

where  $C_{\eta j} = 0$  if the number of edges to be removed from path  $p_j$  equals  $\eta$ , and  $C_{\eta j} > 0$  otherwise. The new penalty term is

$$H'_{pen} = \lambda_p \sum_{j=1}^k \prod_{\eta=1}^{c'_j} C_{\eta j} \tag{14}$$

where  $c'_j = 1$  if  $c_j = 0, 1$ , and  $c'_j = c_j$  if  $c_j > 1$ .

Function  $H'_{pen}$  has as objective to penalize sets of edges of cardinality greater than  $k$  rather than penalizing sets of edges that do not correspond to a multicut as the term  $H_{pen}$  does.  $H'_{pen} = 0$  if and only if every path  $p_j$  is disconnected, and  $H'_{pen} > 0$  otherwise. The degree of  $H'_{pen}$  is at most twice the maximum number of intersections in all paths, i.e.

$$\deg(H'_{pen}) \leq 2 \max \{c'_j | j = 1, \dots, k\} \leq 2(k - 1)$$

which does not depend on the length of paths  $p_i$  in  $G$ .

The penalty function  $H'_{pen}$  can be reduced into a QUBO function using the method in<sup>12</sup>, producing a quadratic expression, the size of which is polynomially bounded in  $\text{size}(H'_{pen})$  and the number of new variables is  $O(n^{2 \log d})$  where  $d = \deg(H'_{pen})$  and  $n = |P|$ . A disadvantage of this method is that the resulting quadratic function has many large coefficients and also introduces many positive quadratic terms. These two effects make the minimization of the resulting function a hard problem<sup>22,23</sup>.

*A Boolean circuit construction for  $H'_{pen}$ .* Let  $\mathbf{x} = \{x_i | 1 \leq i \leq N\}$  be a set of  $N$  Boolean variables, and let  $b$  be a positive integer. Based on section 2.2, the penalty term  $H'_{pen}$  can be written in terms of a pseudo Boolean function  $\phi_{c'_j}$  as follows

$$H'_{pen}(\mathbf{x}) = \lambda_p \sum_{j=1}^k \phi_{c'_j}(\mathbf{x}) \tag{15}$$

satisfying that

$$\phi_{c'_j}(\mathbf{x}) = \begin{cases} 0 & \text{if } l_j - c'_j \leq \sum_{e \in p_j} x_e \leq l_j - 1, \\ \geq b & \text{otherwise.} \end{cases} \quad (16)$$

Function  $\phi_{c'_j}$  can be constructed if we consider a Boolean circuit  $g: \{0, 1\}^N \rightarrow \{0, 1\}$  such that  $g(\mathbf{x}) = 1$  when  $\phi_{c'_j}(\mathbf{x}) > 0$ , and  $g(\mathbf{x}) = 0$  otherwise. This Boolean circuit can always be expressed as a pseudo Boolean function  $\phi_g = \phi_{c'_j}$  such that they both have the same values at every point. Let us formalize this result. A *disjunctive form* (DF) is an expression of the form

$$\varphi = \bigvee_{k=1}^m \left( \bigwedge_{i \in A_k} x_i \wedge \bigwedge_{j \in B_k} \neg x_j \right) \quad (17)$$

where  $A_k \cap B_k = \emptyset$  for  $k = 1, \dots, m$ .

A DF is said to be *orthogonal* if  $(A_k \cap B_l) \cup (A_l \cap B_k) = \emptyset$  for all  $k, l \in \{1, \dots, m\}$  with  $k \neq l$ . An orthogonal DF  $\varphi$  represents a Boolean circuit  $g: \{0, 1\}^N \rightarrow \{0, 1\}$  if the truth value points of  $g$  coincide with the truth value points of  $\varphi$ . Furthermore, every Boolean circuit  $g: \{0, 1\}^N \rightarrow \{0, 1\}$  represented by an orthogonal DF has an associated multilinear polynomial introduced in<sup>24</sup> and presented in (18),

$$\phi_g(\mathbf{x}) = \sum_{k=1}^m \left( \prod_{i \in A_k} x_i \prod_{j \in B_k} (1 - x_j) \right). \quad (18)$$

The Boolean circuit  $g: \{0, 1\}^N \rightarrow \{0, 1\}$  can be constructed using the standard Karnaugh map procedure<sup>25</sup>. This technique receives as input a truth table on  $N$  Boolean variables and returns a Boolean circuit expressed as a sum of products (SOP) of the involved variables which can be implemented using AND, OR and NOT gates. After obtaining the Boolean formulae in terms of the elementary Boolean gates, the NOT gate is interpreted as  $\neg x = 1 - x$ .

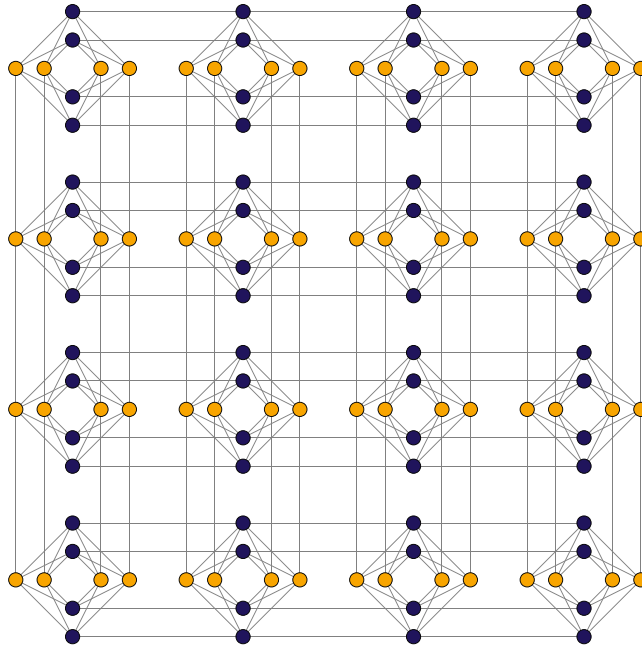
For instance, the corresponding pseudo-Boolean function obtained through Karnaugh maps as SOP can be expressed as

$$\phi_{c'_j}(\mathbf{x}) = \sum_{i=1}^{c'_j} \sum_{A \subseteq p_j, |A|=l_j-i} \prod_{e \in A} (1 - x_e) + \prod_{e \in p_j} x_e. \quad (19)$$

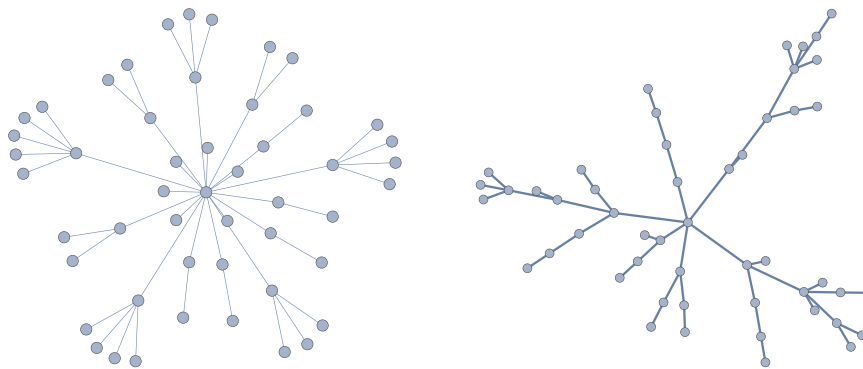
As it can be seen in (19), the degree of function  $\phi_{c'_j}$  is equal to the length of path  $p_j$ . This penalty function has the same degree as the expression given in (8), however, the former has the property that penalizes multicuts of cardinality greater than  $k$ .

**Experiments and Discussion.** We now present our algorithms and corresponding outcomes, as well as the limitations of our proposal. First, let us review the road map (steps) for solving an optimization problem using a quantum annealing approach:

- (i) Select an optimization combinatorial problem (in our case, the MMC problem).
- (ii) Construct a pseudo-Boolean function on binary variables for the selected optimization problem, so that those assignments that minimize the expression also correspond to solutions of the given problem. Since the D-Wave quantum processor has a specific architecture, it only supports pairwise interaction between qubits. Hence, the constructed pseudo-Boolean function must be of degree two, i.e., a QUBO expression. It is not always possible to directly obtain a QUBO expression for a given optimization problem, so in practice we have a high degree pseudo-Boolean function which is to be transformed into a QUBO expression at a later stage.
- (iii) Once we have computed a QUBO expression for the selected problem, the next step is to embed the logical problem into the fixed architecture of the quantum processor. This architecture is represented as a Chimera graph which has a limited interconnectivity (see Fig. 1 for an example of a Chimera graph). At this step, a key feature to remember is physical resources vs logical variables as, in order to embed a logical problem with arbitrary interconnectivity, we will frequently need a larger number of physical qubits than the number of logical variables; moreover, if the number of logical variables scales up rapidly, we may run out of physical qubits. The estimation of the minimum number of physical qubits required to embed a logical problem is an active area of research<sup>26,27</sup>. The Chimera graph architecture implements an Ising model, hence the embedding process maps logical variables into Ising variables. Another important consideration is that the D-Wave processor has limited precision to represent Ising coefficients; consequently, the coefficients of the QUBO expression should not be too large so that they can be correctly mapped into the quantum processor and results can be discriminated from noise level.
- (iv) Finally, the annealing process is initiated to find the minimum energy configuration in which the solution of the problem is codified.



**Figure 1.** Chimera graph topology of  $4 \times 4$  blocks.



**Figure 2.** Example random trees: (left) using the Erdős-Rényi model for  $n = 50$  and  $p = 0.3$ , and (right) using the Watts-Strogatz model with  $n = 50$ ,  $r = 2$  and  $\beta = 0.12$ .

For steps (i) and (ii), we consider the QUBO formulations of the MMC problem presented in section 2: the formulation with direct reduction, the one based on crossing of paths and the one based on logical circuits. For the last two cases, the reduction methods cited in Section 1.1 are used to obtain the QUBO functions.

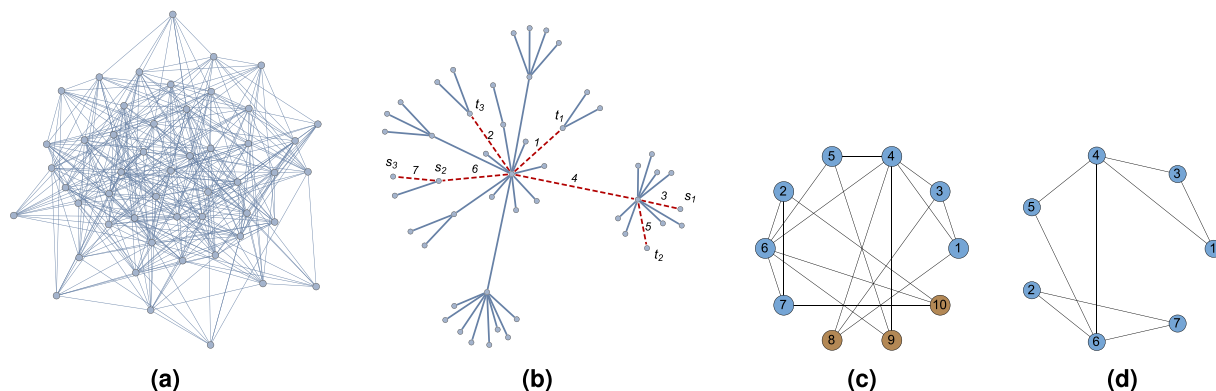
In the following section, we consider the mapping problem presented above in step (iii), namely mapping logical variables in a QUBO function to physical qubits in the actual hardware architecture.

**Minor embedding.** The D-Wave processor can be represented by an architecture known as a *Chimera graph* which consists of an  $M \times N$ -lattice of blocks each one having  $2L$  physical qubits for a total of  $2MNL$  qubits (see Fig. 1). Each block in the Chimera graph is a  $L$ -bipartite graph and each physical qubit is connected with at most six other qubits. To solve a problem using a D-Wave processor, it is necessary to represent an Ising/QUBO problem as a subgraph of the Chimera. However, it is seldom possible to find a one-to-one mapping of logical variables with physical qubits.

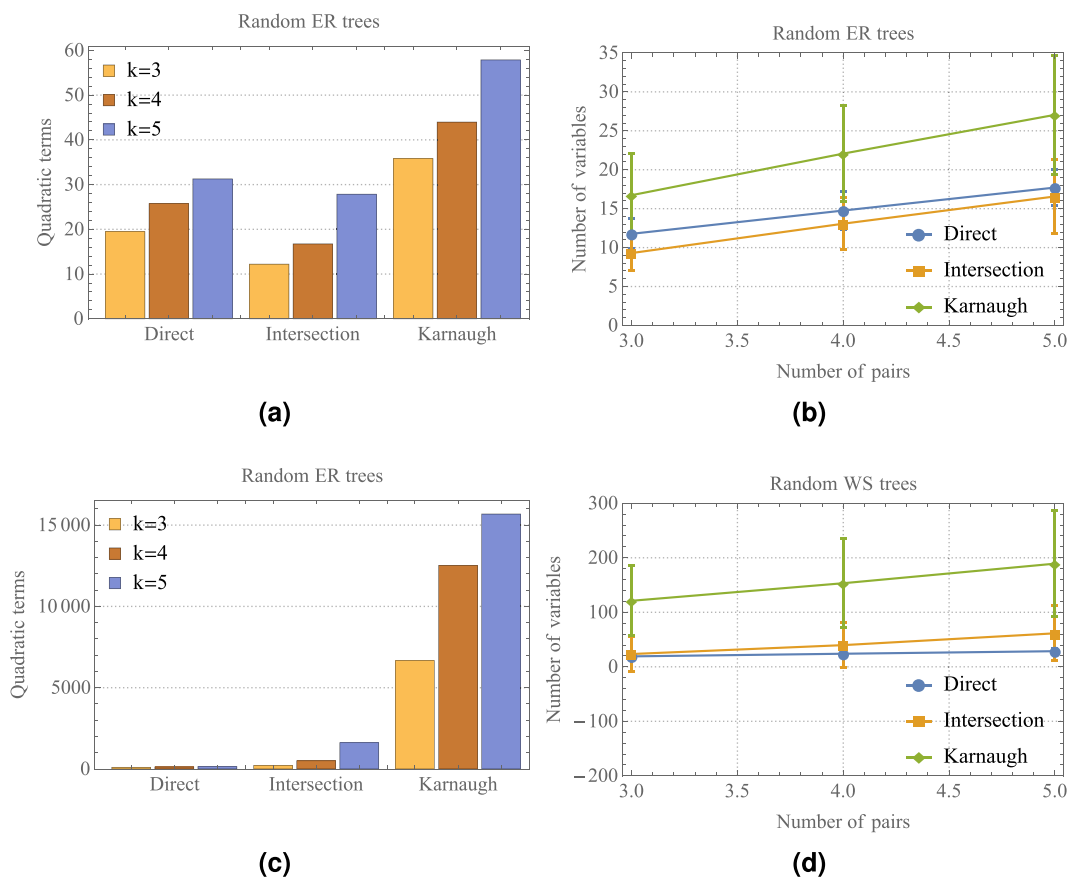
The method to find an equivalent subgraph into the Chimera to a given logical problem expressed as an Ising/QUBO function is called *minor embedding* and it is stated as follows:

**Problem 2:(Minor embedding)** Given a Chimera graph  $\mathcal{G}_{M,N,L}$  and a logical graph  $G = (V, E)$ , find a subgraph in  $\mathcal{G}_{M,N,L}$  such that

1. Each vertex  $j \in V$  is mapped to a connected subtree  $T_j$  in  $\mathcal{G}_{M,N,L}$ .
2. Each edge  $\{i, j\} \in E$  must be mapped to at least one coupler in  $\mathcal{G}_{M,N,L}$ .



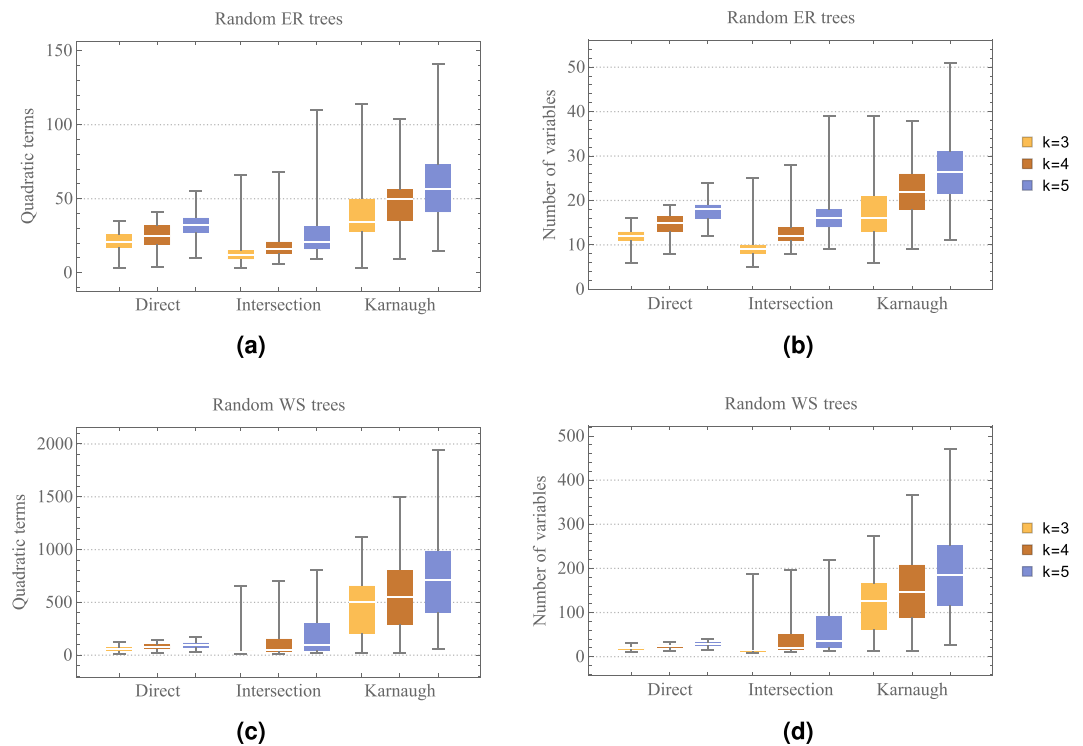
**Figure 3.** (a) ER random graph  $G(50, 0.3)$ , (b) corresponding spanning tree, (c) logical graph of the QUBO reduction of functions (20) and (22), and (d) logical graph of the QUBO expression given in (21).



**Figure 4.** Comparison of the number of quadratic terms generated using the direct, intersection and Karnaugh methods.

Finding a minor embedding which uses the minimum number of qubits is a hard problem in general<sup>26</sup>. However, there are heuristic algorithms that obtain approximated minor embeddings in polynomial time. One of these techniques is the one proposed in<sup>28</sup> which maps a logical variable to a chain of qubits. The main disadvantage of these techniques is that they create long chains of qubits to allow the connectivity of the logical problem into the Chimera.

In this paper, we use the D-Wave solver application programming interface (SAPI) which provides heuristic algorithms for minor embedding. Heuristic algorithms are constrained by a peculiar property: their time complexity (i.e., their worst and average runtime) is usually lower than exhaustive search algorithms at the cost of producing approximative solutions most of the time and optimal results only rarely. In our case, SAPI heuristic algorithms produce different embeddings that are not always optimal with respect to the number of physical qubits. For each of our experiments, we have run the heuristic embedding algorithms twenty times and have



**Figure 5.** Descriptive comparison of the number of quadratic terms and number of variables generated using the direct, intersection and Karnaugh methods.

chosen only those embeddings that have shorter qubit chains. This process is carried out on a conventional digital computer, before the adiabatic process.

**Instance problem generation.** For this paper, we have generated random instances of the MMC problem, namely random trees, according to the following procedure: (1) generate a connected random graph<sup>16</sup>, (2) compute their corresponding spanning tree (ST) and (3) assign integer positive weights to the edges of the obtained ST. Here, we consider the generation of connected random graphs using the Erdős-Rényi (ER) and the Watts-Strogatz (WS) models. In the ER model, a random graph  $G(n, p)$  with  $n$  vertices is constructed by including each possible edge with a probability  $p$  independently from every other edge. In the WS model, a random graph  $G(n, r, \beta)$  with  $n$  vertices is constructed by creating an initial ring of vertices with each vertex connected to its  $r$  nearest neighbors. Thus, replace every possible edge  $\{u, v\}$  with a new edge  $\{u, v\}$  with probability  $\beta$ , duplicated edges are forbidden, but original edges may end up being reinstated.

Figure 2 shows some examples of trees generated using the ER and WS models. We chose the ER and WS models because they allow us to generate trees with small and large diameters, respectively. For instance, Fig. 2 (left) shows a tree with a diameter of 4 using the ER model and Fig. 2 (right) shows a tree with a diameter of 10 using the WS model, in both cases the number of vertices in the random graph are the same.

Let us consider an example of how the QUBO formulations given in Section 2 for the MMC problem are constructed. Figure 3(a) shows a random ER graph for 50 vertices and probability  $p = 0.3$ , and Fig. 3(b) shows its corresponding random spanning tree. Assume an instance of the MMC problem with unitary weights for  $k = 3$  and set of pairs  $\{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}$ ; from Fig. 3(b), it can be seen that paths  $p_j, j = 1, 2, 3$  contain edges  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$  and  $\{2, 6, 7\}$ , respectively. Notice that the length of paths  $p_j$  is  $l_j = 3$  for  $j = 1, 2, 3$ , and paths  $p_1$  and  $p_2$  share the edge 4, and paths  $p_2$  and  $p_3$  share the edge 6.

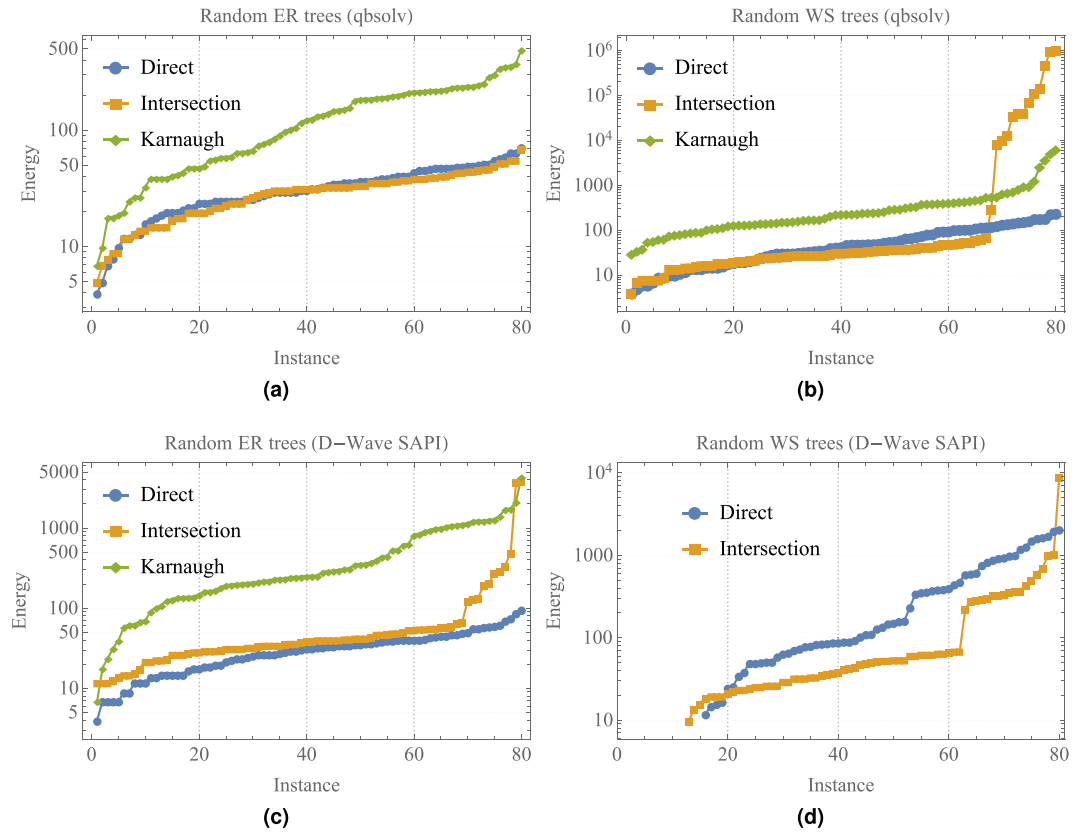
The direct mapping given in (6) for the instance of the MMC problem in Fig. 3(b) is

$$H = 7 - x_1 - x_2 - x_3 - x_4 - x_5 - x_6 - x_7 + 7x_3x_4x_1 + 7x_4x_5x_6 + 7x_2x_6x_7. \quad (20)$$

The logical graph obtained from the QUBO reduction of (20) can be seen in Fig. 3(c) where the blue color vertices are the original problem variables and the brown color vertices are the introduced new variables. Notice that the induced subgraphs from vertices  $\{1, 3, 4\}$  and  $\{4, 5, 6\}$  share vertex 4, and the induced subgraphs from vertices  $\{4, 5, 6\}$  and  $\{2, 6, 7\}$  share vertex 6. Also, notice that vertices  $\{1, 3, 4\}$  together with the new variable 8 form a complete graph; the same is also true for vertices  $\{4, 5, 6\}$  and  $\{2, 6, 7\}$  together for new variables 9 and 10, respectively. From the above, it can be concluded that if the paths do not intersect each other, then their corresponding logical graphs will be disconnected.

On the other hand, the mapping based on crossing paths given in (14) for the instance of the MMC problem in Fig. 3(b) is





**Figure 6.** Sorted energy solutions obtained using the `qbsolv` and D-Wave simulator over 80 random instances using the direct, intersection and Karnaugh QUBO constructions.

$$\begin{aligned}
 H = & 91 - 22x_1 - 22x_2 - 22x_3 - 43x_4 - 22x_5 - 43x_6 - 22x_7 + 14x_3x_1 + 14x_4x_1 \\
 & + 14x_3x_4 + 14x_4x_5 + 14x_2x_6 + 14x_4x_6 + 14x_5x_6 + 14x_2x_7 + 14x_6x_7
 \end{aligned} \tag{21}$$

which is already in QUBO form since we penalize cuts of cardinality at most three. In other words, each path  $p_j$ ,  $j = 1, 2, 3$  will be disconnected by removing at most one edge. Figure 3(d) shows the logical graph of expression (21) where no new variables are used. Notice that only the induced subgraphs of vertices  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$  and  $\{2, 6, 7\}$  are shown.

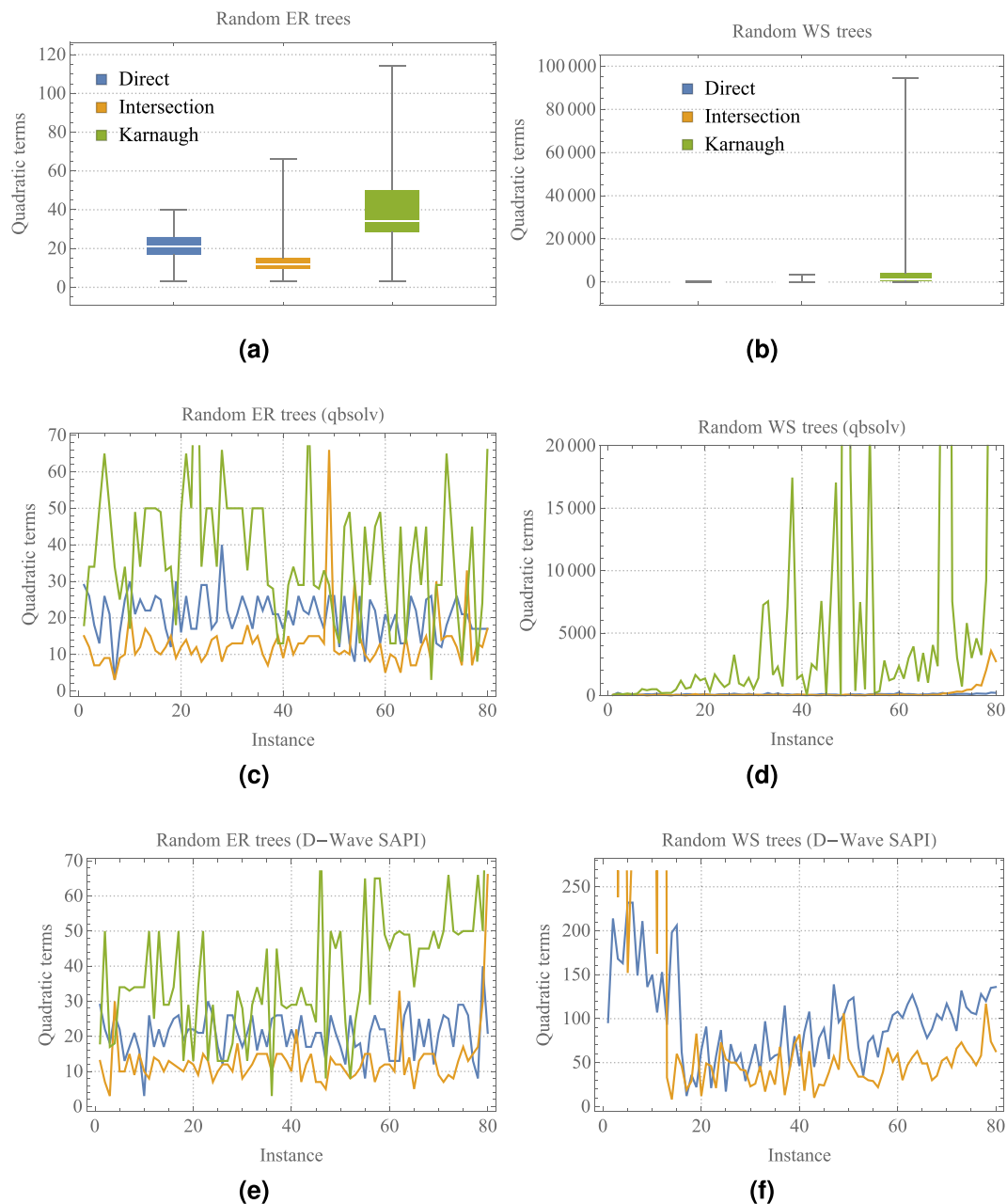
Finally, the Karnaugh mapping given in (19) for the instance of the MMC problem in Fig. 3(b) is

$$\begin{aligned}
 H = & 70 - 15x_1 - 15x_2 - 15x_3 - 29x_4 - 15x_5 - 29x_6 - 15x_7 + 7x_3x_1 \\
 & + 7x_4x_1 + 7x_3x_4 + 7x_4x_5 + 7x_2x_6 + 7x_4x_6 + 7x_5x_6 + 7x_2x_7 \\
 & + 7x_6x_7 + 7x_3x_4x_1 + 7x_4x_5x_6 + 7x_2x_6x_7
 \end{aligned} \tag{22}$$

whose logical graph coincides with the QUBO reduction of (20).

**Problem scaling.** Figure 4 shows a comparison of the number of quadratic terms generated by the proposed QUBO formulations presented in sections 2.1, 2.2 and 2.3, that will be named as the direct, intersection and Karnaugh methods, respectively. We generated random trees using the ER and WS models with probabilities  $p = 0.3$  and  $\beta = 0.12$ , for  $n = 50$  and number of pairs  $k = 3, 4, 5$ . We generate random WS trees with paths between the pairs  $(s_i, t_i)$  of length less than 8. For each random tree model used, the corresponding QUBO function was constructed using the direct, intersection or Karnaugh methods. The energy function given in (6) was used for the three methods with different penalty terms. The direct method corresponds to the penalty term as presented in (8), the intersection method has as penalty term the expression shown in (14), and the Karnaugh method has as penalty term the expression given in (19). We use the reduction method of Ishikawa<sup>14</sup> explained in section 2.1 as well as Freedman’s method<sup>13</sup>, to obtain the corresponding QUBO function.

Figure 4(a) presents is the average number of quadratic terms over 100 ER random trees, generated using the direct, intersection and Karnaugh methods for  $k = 3, 4, 5$ . Figure 4(b) shows their corresponding average number of variables after the degree reduction of the constructed QUBO functions shown in Fig. 4(a). The vertical lines in Fig. 4(b,d) indicate a standard deviation. As it can be seen in the Fig. 4(a,b), the intersection method uses the minimum number of variables to represent the MMC problem and the Karnaugh method requires more variables to represent the same problem. It is important to mention that the average diameter of the generated ER random trees was 4. We choose  $n = 50$  since the number of quadratic terms is an increasing function with respect to  $n$ . Similarly,



**Figure 7.** Visualisation of the number of quadratic terms based on the sorted energies shown in Fig. 6 over 80 random instances using the direct, intersection and Karnaugh QUBO constructions.

Fig. 4(c,d) show the average number of quadratic terms using WS random trees for the direct, intersection and Karnaugh methods. It is remarkable that Karnaugh's method requires a huge number of quadratic terms and variables in comparison with the direct and intersection methods. On the other hand, the direct method uses the minimum number of quadratic terms. In this case, the average diameter of the generated WS random trees was 10.

Figure 5 presents an even more descriptive comparison of the number of quadratic terms and number of variables obtained after degree reduction, for the direct, intersection and Karnaugh methods. The area in each box represents the interquartile range, the statistical median as a horizontal line in the box, and its vertical lines the lower and upper whiskers. The data shown in Fig. 5(a–d) correspond to the same data shown in Fig. 4. As can be noted from Fig. 5, the lower and upper whiskers for the Karnaugh method are distant from their median, which explains the standard deviation in Fig. 4(b,d).

In view of these scaling results, it can be seen that Karnaugh's construction given in (19) has terms of degree equal to the length of the paths between pairs  $(s_i, t_i)$ . Thus, when applying a reduction method on (19), the number of quadratic terms increases, as opposed to the method given in (14) whose degree depends on the number of crossing paths. As a consequence, the dimension of the search space of its corresponding optimization problems also increases.

**Classical solution performance.** In order to test the performance of obtaining solutions using the proposed formulations, the qbsolv tool was used to solve large QUBO problems by partitioning into subproblems targeted for execution on a D-Wave system<sup>29</sup>. The qbsolv tool returns approximated solutions of large QUBO problems. The experiments were performed on a Desktop computer MacBook Air with a Intel Core i5 processor at 1.3 GHz and 4 GB of RAM. We generated 80 random trees for  $n = 50$  and  $k = 3$  using the ER and WS models, and for each random tree, their corresponding QUBO functions were constructed using the direct, intersection and Karnaugh methods.

Before mapping a QUBO expression to Ising, constant terms are omitted without changing the original problem because constant terms cannot be represented in the graph topology. For the D-Wave SAPI software, an auto-scaling mode is used which utilizes the largest possible parameter ranges. In the case of the qbsolv tool we do not have precision limitations on the parameters, so each QUBO instance is directly submitted.

Figure 6(a,b) show the energy solutions obtained using qbsolv for ER and WS random trees, respectively. To compare the approximated energies solutions, they were sorted in an increasing order over the same set of instances for the direct, intersection and Karnaugh methods. Figure 6(a,b) show that the direct and intersection methods have similar energy solutions for the ER and WS random trees. On the other hand, the Karnaugh method has larger energy solutions. The latter can be explained since the large number of used variables in the Karnaugh method, as can be seen in Fig. 4. We were unable to calculate the optimal energy solutions of the random instances since the large number of variables. The cases for  $k = 4, 5$  were omitted since they have similar behavior.

The D-Wave SAPI also provides classical algorithms to solve QUBO/Ising problems by using simulated quantum annealing (SQA)<sup>3</sup> on a Chimera graph of dimension  $4 \times 4$  blocks. Figure 6(c,d) show the energy solutions obtained using D-Wave SAPI for ER and WS random trees, respectively. In this case, we use the same 80 random trees for  $n = 50$  and  $k = 3$  as in Fig. 6(a,b). For each instance of the MMC problem, 1000 readouts was requested to the SQA algorithm. Figure 6(c) shows the minimum energies among the requested 1000 readout per instance, for ER random trees. As can be seen in Fig. 6(c), the obtained energies have a similar behavior as in Fig. 6(a). Figure 6(a) also shows the energies for the case of WS random trees using the SQA algorithm. In this latter case, it was not possible to obtain the energies for the Karnaugh method since the embedding algorithm fails to embed large QUBO functions in a Chimera graph of a  $4 \times 4$  dimension.

Figure 7 shows how the sorted energies in Fig. 6 are correlated with respect to the number of quadratic terms in the set of instances. Figure 7(a,b) present the distribution of quadratic terms for the direct, intersection and Karnaugh methods using random ER and WS trees in Fig. 6. Also, Fig. 7(c–f) show the number of quadratic terms of the direct, intersection and Karnaugh methods, where the instances were arranged based on the same energy sorting shown in Fig. 6(a–d), respectively. For instance, the higher energies in Fig. 6(a) are correlated with the higher number of quadratic terms generated by the Karnaugh method. The same correlation is observed for the classical solvers qbsolv and D-Wave simulator.

We have proposed QUBO formulations of the MMC problem restricted to the family of trees. Our simulation results show that the direct and intersection methods have a similar scaling, in the number of quadratic terms, and performance. On the other hand, the Karnaugh method has the worst results, mainly by the required large number of variables and quadratic terms in its construction.

## Discussion

We have proposed a method to formulate the Minimum Multicut Problem into the QUBO representation, and the technical difficulties faced when embedding and submitting a problem to the quantum annealer processor. We have considered a special NP – hard case of the Minimum Multicut problem based on random trees representations. This special case allows us to formulate a QUBO expression that can be embedded into the Chimera graph using a moderate number of qubits after the degree reduction of the high degree expression. Moreover, we have proposed the Karnaugh method to analytically construct instances of the MMC problem and the preliminary results of our algorithms are promising.

Received: 30 June 2019; Accepted: 31 October 2019;

Published online: 20 November 2019

## References

- Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680, <https://doi.org/10.1126/science.220.4598.671> (1983).
- Finnila, A., Gomez, M., Sebenik, C., Stenson, C. & Doll, J. Quantum annealing: A new method for minimizing multidimensional functions. *Chem. Phys. Lett.* **219**, 343–348, [https://doi.org/10.1016/0009-2614\(94\)00117-0](https://doi.org/10.1016/0009-2614(94)00117-0) (1994).
- Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **58**, 5355–5363, <https://doi.org/10.1103/PhysRevE.58.5355> (1998).
- Brooke, J., Bitko, D., Rosenbaum, T. F. & Aeppli, G. Quantum Annealing of a Disordered Magnet. *Science* **284**, 779, <https://doi.org/10.1126/science.284.5415.779> (1999).
- Farhi, E. *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475, <https://doi.org/10.1126/science.1057726> (2001).
- Santoro, G. E. & Tosatti, E. Optimization using quantum mechanics: quantum annealing through adiabatic evolution. *J. Phys. A: Math. Gen.* **39**, R393–R431, <https://doi.org/10.1088/0305-4470/39/36/r01> (2006).
- Das, A. & Chakrabarti, B. K. *Colloquium: Quantum annealing and analog quantum computation.* *Rev. Mod. Phys.* **80**, 1061–1081, <https://doi.org/10.1103/RevModPhys.80.1061> (2008).
- Perdomo, A., Truncik, C., Tubert-Brohman, I., Rose, G. & Aspuru-Guzik, A. Construction of model hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models. *Phys. Rev. A* **78**, 012320, <https://doi.org/10.1103/PhysRevA.78.012320> (2008).
- Perdomo-Ortiz, A., Fluegemann, J., Narasimhan, S., Biswas, R. & Smelyanskiy, V. A quantum annealing approach for fault detection and diagnosis of graph-based systems. *The Eur. Phys. J. Special Top.* **224**, 131–148, <https://doi.org/10.1140/epjst/e2015-02347-y> (2015).

10. McGeoch, C. C. & Wang, C. Experimental evaluation of an adiabatic quantum system for combinatorial optimization. In *Proceedings of the ACM International Conference on Computing Frontiers, CF' 13*, 23:1–23:11, <https://doi.org/10.1145/2482767.2482797> (ACM, New York, NY, USA, 2013).
11. Barahona, F. On the computational complexity of ising spin glass models. *J. Phys. A: Math. Gen.* **15**, 3241 (1982).
12. Boros, E. & Hammer, P. L. Pseudo-boolean optimization. *Discret. Appl. Math.* **123**, 155–225, [https://doi.org/10.1016/S0166-218X\(01\)00341-9](https://doi.org/10.1016/S0166-218X(01)00341-9) (2002).
13. Freedman, D. & Drineas, P. Energy minimization via graph cuts: Settling what is possible. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 20–26 June 2005, San Diego, CA, USA, 939–946, <https://doi.org/10.1109/CVPR.2005.143> (2005).
14. Ishikawa, H. Transformation of general binary mrf minimization to the first-order case. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 1234–1249, <https://doi.org/10.1109/TPAMI.2010.91> (2011).
15. Venegas-Andraca, S., Cruz-Santos, W., McGeoch, C. & Lanzagorta, M. A cross-disciplinary introduction to quantum annealing-based algorithms. *Contemp. Phys.* **59**, 174–197, <https://doi.org/10.1080/00107514.2018.1450720> (2018).
16. Bollobás, B. *Graph theory: an introductory course*. Graduate texts in mathematics (Springer Verlag, 1979).
17. Yannakakis, M., Kanellakis, P., Cosmadakis, S. & Papadimitriou, C. Cutting and partitioning a graph after a fixed pattern. In Diaz, J. (ed.) *Automata, Languages and Programming, vol. 154 of Lecture Notes in Computer Science*, 712–722, <https://doi.org/10.1007/BFb0036950> (Springer Berlin Heidelberg, 1983).
18. Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D. & Yannakakis, M. The complexity of multiterminal cuts. *SIAM J. Comput.* **23**, 864–894, <https://doi.org/10.1137/S009753979225297> (1994).
19. Garg, N., Vazirani, V. & Yannakakis, M. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica* **18**, 3–20, <https://doi.org/10.1007/BF02523685> (1997).
20. Stone, H. S. Multiprocessor scheduling with the aid of network flow algorithms. *IEEE Transactions on Softw. Eng.* **3**, 85–93 (1977).
21. Brunetta, L., Conforti, M. & Fischetti, M. A polyhedral approach to an integer multicommodity flow problem. *Discret. Appl. Math.* **101**, 13–36, [https://doi.org/10.1016/S0166-218X\(99\)00187-0](https://doi.org/10.1016/S0166-218X(99)00187-0) (2000).
22. Boros, E. & Gruber, A. On quadratization of pseudo-boolean functions. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2012, Fort Lauderdale, Florida, USA, January 9–11* (2012).
23. Boros, E., Crama, Y. & Rodríguez-Heck, E. Quadratisations of symmetric pseudo-boolean functions: sub-linear bounds on the number of auxiliary variables. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2018, Fort Lauderdale, Florida, USA, January 3–5* (2018).
24. Crama, Y. & Hammer, P. *Boolean Functions: Theory, Algorithms, and Applications*. Encyclopedia of Mathematics and its Applications (Cambridge University Press, 2011).
25. Karnaugh, M. The map method for synthesis of combinational logic circuits. *Transactions Am. Inst. Electr. Eng. Part I: Commun. Electron.* **72**, 593–599, <https://doi.org/10.1109/TCE.1953.6371932> (1953).
26. Choi, V. Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design. *Quantum Inf. Process.* **10**, 343–353, <https://doi.org/10.1007/s11128-010-0200-3> (2011).
27. Bian, Z. *et al.* Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Front. ICT* **3**, 14, <https://doi.org/10.3389/fict.2016.00014> (2016).
28. Cai, J., Macready, W. G. & Roy, A. A practical heuristic for finding graph minors. *ArXiv e-prints*, 1406.2741 (2014).
29. Booth, M., Reinhardt, S. P. & Roy, A. Partitioning Optimization Problems for Hybrid Classical/Quantum Execution. *Tech. Rep.*, D-Wave The Quantum Computing Company (2017).

## Acknowledgements

S.E.V.-A. warmly thanks his family for their unconditional support and acknowledges the financial support of Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias and CONACyT (SNI Number 41594 as well as Fronteras de la Ciencia project No. 1007). M.L. acknowledges the support received by the U.S. Naval Research Laboratory Naval Innovative Science & Engineering (NISE) program.

## Author contributions

W.C.S. proposed the idea of writing QUBO formulations for the MCC problem on tree graphs. W.C., S.E.V.-A. and M.L. made contributions towards writing and simulating QUBO expressions and algorithms presented in this paper. W.C., S.E.V.-A. and M.L. reviewed the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to S.E.V.-A.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019