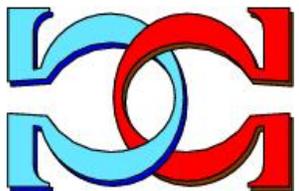
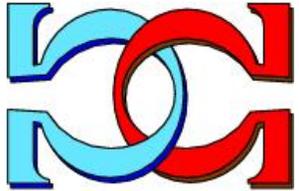
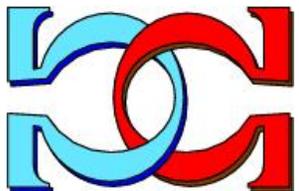


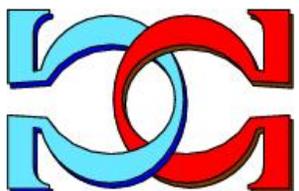
**CDMTCS
Research
Report
Series**



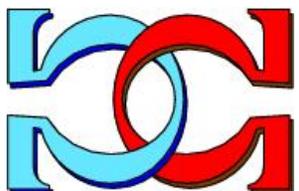
**An Equivalent QUBO Model
to the Minimum Multi-Way
Cut Problem**



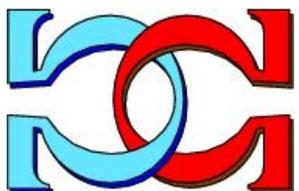
**Shahrokh Heidari
Michael J. Dinneen
Patrice Delmas**



Department of Computer Science,
University of Auckland,
Auckland, New Zealand



CDMTCS-565
September 2022



Centre for Discrete Mathematics and
Theoretical Computer Science

An Equivalent QUBO Model to the Minimum Multi-Way Cut Problem

Shahrokh Heidari^{1,2} Michael J. Dinneen^{1*} Patrice Delmas²

¹School of Computer Science, The University of Auckland, New Zealand

²Intelligent Vision Systems Lab, The University of Auckland, New Zealand

Abstract

Motivated by an application in Computer Vision, we present an efficient QUBO solution for the *minimum multi-way cut* problem. For an edge-weighted input graph $G = (V, E)$ and a set of terminals $T = \{t_1, t_2, \dots, t_k\} \subset V$ we want to find a subset of edges E_c of minimum total cost such that $G' = G \setminus E_c$ separates T . QUBO representations are useful for solving problems on an adiabatic quantum computer like those produced by D-Wave Systems. Our reduction from the multi-way cut problem requires only $O(k|V|)$ binary variables/qubits. The main result of this paper is a proof of correctness of this model. Furthermore, our reduction is small enough to be able to empirically test it with an existing D-Wave hybrid quantum-classical solver, which is illustrated at the end of this paper.

Keywords: Multi-way cut, Quantum Annealing, D-Wave, QUBO, Computer Vision, Image Restoration.

1 Introduction

Generally, the *minimum multicut* problem is defined as follows. Let $G(V, E, C)$ be an arbitrary undirected weighted graph, where V is the set of vertices, E is the set of edges, and $C : E \rightarrow \mathbb{R}$ is a weight function on the edges. Let $(s_1, t_1), \dots, (s_k, t_k)$ be a collection of vertex pairs in V . The *minimum multicut* problem is to find a subset of edges with minimum total weight whose removal separates s_i from t_i for $1 \leq i \leq k$ [2]. A special case of the *minimum multicut* problem is the *minimum multi-way cut* problem. Let $T = \{t_1, t_2, \dots, t_k\} \subset V$ be a set of terminals. A multi-way cut on G along with the set of terminals T is a subset of edges $E_c \subset E$ whose removal disconnects all the terminals in $G(V, E/E_c)$. The cost of a multi-way cut is the sum of the weights of its edges, and the *minimum multi-way cut* problem is to find a multi-way cut with the minimum cost [8]. This problem is not particularly new and has

*Contact author: mjd@cs.auckland.ac.nz

been studied for many years for a wide range of applications such as Very Large Scale Integration (VLSI) system design, Parallel Computing, Distributed Computing, Clustering [27], and Computer Vision [3, 24]. For $k = 2$, the *minimum multi-way cut* problem is referred as the minimum cut problem which can be solved in polynomial time by flow algorithms such as Ford-Folkerson [14], Dinic [11], and Push-relabel [15] algorithms (see [25] for the recent updates). For $k \geq 3$, the *minimum multi-way cut* problem is NP-Hard [8] for which several approximation algorithms have been proposed [4, 7, 22, 29] (see [27] for more information).

Despite having a long tradition, most of the research on the minimum multi-way cut problem is aimed at approximating the optimal solution. With the advent of quantum computations, recent studies have focused on leveraging quantum properties to possibly overcome intractable classical problems. Quantum computers are known to have potentially lower-time complexity on certain problems than the best-classical counterparts [9, 16, 28]. D-Wave system was the first company to build a quantum computer that realizes the quantum Ising-spin glass Hamiltonian on a special graph. The importance of Ising models is because a variety of NP-Hard optimization problems can be solved by finding the ground state of the corresponding Ising models [19]. D-Wave Quantum Processing Units (QPUs) naturally approximate the ground state of an Ising model, and they have provided efficient solutions for different instances of NP-hard problems [5, 6, 21]. Therefore, it would be of special interest to have a quantum model to solve the minimum multi-way cut problem. In 2019, Cruz-Santos et al. proposed two quantum models to solve the *minimum multicut* problem on a family of connected trees. However, to the best of our knowledge, no study has so far considered a quantum model to solve the *minimum multi-way cut* problem. The main objective of this study is to investigate a quantum model to solve this NP-Hard problem for arbitrary weighted graphs.

The rest of the paper is organized as follows: In Section 2, we provide a brief introduction to Quantum Annealing computations. An equivalent quantum model to the *minimum multi-way cut* and its proof of correctness are given in Section 3. In Section 4, we show an application of the *minimum multi-way cut* problem in Computer Vision using the proposed quantum model. Finally, we conclude with some comments and open problems in Section 5.

2 Quantum Annealing

The first universal quantum model of computation was the Quantum Gate model developed by Deutsch [10]. The main goal was to leverage quantum mechanical properties to show a quantum speedup over the classical computation. In this model, a collection of quantum logic gates, called quantum circuits, can compute any classical function [10]. An alternative equivalent [1] to the Quantum Gate model is the Quantum Annealing model, which was introduced by Farhi et al. [13]. In this model, quantum bits (qubits) are particles in a quantum dynamical system that evolve over time based on special forces acting on them. These forces are some sorts of constraints which are either external (from other sources) or internal (from interactions among qubits). Each state of a register of n qubits $\{0, 1\}^n$ has

an energy based on the applied forces. A time-dependent Hamiltonian is a mathematical description of a system that gives the energy of the system and characterizes the forces at any time [21]. Quantum Annealing is the process of finding a state of the system that has the lowest energy based on the time-dependent Hamiltonian. Therefore, a Quantum Annealing algorithm is to solve an optimization problem that is based on an objective function (see [21] for more information). Practically, a quantum annealer (such as a D-Wave QPU) is needed to accomplish this process. To prepare the objective function for the minimization by a D-Wave QPU, it should be formulated as either Ising model or Quadratic Unconstrained Binary Optimization (QUBO) model. Since there is a simple transition between these two models [21], we only explain the latter. Given a column n -vector binary variables $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$, a QUBO model is written as $q(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$, where \mathbf{Q} is an $n \times n$ matrix that can be chosen to be upper-diagonal. Therefore, $q(\mathbf{x})$ can be reformulated as (1).

$$q(\mathbf{x}) = \sum_{i \leq j} \mathbf{Q}_{i,j} x_i x_j = \sum_i \mathbf{Q}_{i,i} x_i + \sum_{i < j} \mathbf{Q}_{i,j} x_i x_j. \quad (1)$$

The diagonal terms $\mathbf{Q}_{i,i}$ are the linear coefficients acting as the external forces (note $x_i^2 = x_i$), and the off-diagonal terms $\mathbf{Q}_{i,j}$ are the quadratic coefficients for the internal forces (e.g. $x_i x_j$ for $i < j$) [21].

A D-Wave QPU is a collection of tiny metal loops accommodated on a special graph. These loops are either physical qubits or couplers. The external and internal forces are applied to the physical qubits and couplers, respectively, as magnetic fields. Each binary variable x_i is called a logical qubit, and it is embedded into the D-Wave QPU graph using an embedding algorithm (several physical qubits could be chained together to represent a single logical qubit in a QUBO model).

3 The minimum multi-way cut problem as a QUBO

In this section, we present an equivalent QUBO model to the *minimum multi-way cut* problem and prove its correctness.

Definition 1. *Given an edge-weighted graph G along with a set of terminals $T = \{t_1, t_2, \dots, t_k\}$, the minimum multi-way cut problem is to find a subset of edges of minimum total weight whose removal separates t_i from t_j for $1 \leq i < j \leq k$ [2].*

If G is not connected, it is sufficient to solve the *minimum multi-way cut* problem on the components of G that contain at least two terminals of T . Therefore, without loss of generality, we assume that G is a connected graph for the rest of the paper. Generally, the *minimum multi-way cut* problem can be solved by a labeling problem [20, p. 6] as follows: A valid multi-way cut corresponds to a labeling $\mathcal{L} : V \rightarrow T$ such that for each $t \in T$, $\mathcal{L}(t) = t$. In this case, $E_c = \{\{u, v\} \mid \mathcal{L}(u) \neq \mathcal{L}(v)\}$ is a multi-way cut, and its cost is computed by $\sum_{\{u,v\} \in E_c} C(\{u, v\})$. Therefore, the *minimum multi-way cut* problem is to find a labeling such that its corresponding valid multi-way cut has the minimum cost. In the following, we partition the set of all vertices V (which also includes k terminal vertices) into k disjoint sets

by a labeling such that each set of vertices including a terminal $t \in T$ is labeled uniquely by t . Any edge $(u, v) \in E$ with u and v in different partition labels means we are deleting that edge for the multi-way cut. Minimizing the sum of the costs of these deleted edges is our optimization problem to find the *minimum multi-way cut*.

Initially, we allocate a set of k binary variables to each $u \in V$. Let $\mathbf{x} \in \{0, 1\}^{k|V|}$ be a set of $k|V|$ binary variables as $\mathbf{x} = \{x_{u,t} \mid u \in V, t \in T\}$, where $|V|$ is the number of vertices. An equivalent QUBO to the *minimum multi-way cut* problem can be defined as $H_{qubo} : \{0, 1\}^{k|V|} \rightarrow \mathbb{R}$ given in (2).

$$H_{qubo}(\mathbf{x}) = \alpha \left(\sum_{u \in V} \left(1 - \sum_{t \in T} x_{u,t} \right)^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} x_{t,t'} \right) \quad (2)$$

$$+ \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t} x_{v,t'},$$

where $\alpha > \sum_{\{u,v\} \in E} C(\{u, v\})$. Let $\mathbf{x}^* = \min_{\mathbf{x}} H_{qubo}(\mathbf{x})$. The set of edges

$$E_m = \{\{u, v\} \mid x_{u,t}^* = x_{v,t'}^* = 1, \{u, v\} \in E, t, t' \in T, \text{ and } t \neq t'\}$$

is the *minimum multi-way cut* on G , and $H_{qubo}(\mathbf{x}^*)$ is the minimum multi-way-cut cost.

3.1 Proof of correctness

The formulation H_{qubo} in (2) has two parts. The first part guarantees each vertex is assigned a unique label from T , and the allocated label to each $t \in T$ is t . The second part is used to calculate the weight of the multi-way cut (the part that we want to optimize).

We start with the first part. For each $u \in V$, we have allocated a set of k binary variables as $\{x_{u,t_1}, \dots, x_{u,t_k}\}$ where if $x_{u,t_i} = 1$ for $1 \leq i \leq k$, then the allocated label to u is t_i . Therefore, we need to make sure that for each $u \in V$, $\{x_{u,t_1}, \dots, x_{u,t_k}\}$ has only one value of 1 across all $\{x_{u,t_i}\}$ for $1 \leq i \leq k$. In other words, $\sum_{t \in T} x_{u,t} = 1$.

Lemma 1. $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 = 0$ if and only if $\sum_{t \in T} x_{u,t} = 1$ for all $u \in V$.

Proof. (\Rightarrow) Suppose $\sum_{t \in T} x_{u,t} = 1$ for all $u \in V$. Then,

$$\sum_{u \in V} \left(1 - \sum_{t \in T} x_{u,t} \right)^2 = |V| (1 - 1)^2 = 0.$$

(\Leftarrow) Now, suppose that $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 = 0$. Since $(1 - \sum_{t \in T} x_{u,t})^2$ is non-negative for all $u \in V$, to have $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 = 0$, the term $1 - \sum_{t \in T} x_{u,t}$ must be equal to zero. Therefore, $\sum_{t \in T} x_{u,t} = 1$. \square

Next, we want to make sure that a terminal vertex $t \in T$ is labeled as t .

Definition 2. \mathbf{x} is said to be a labeling if

(i) for each $u \in V$, $\sum_{t \in T} x_{u,t} = 1$;

(ii) for each $t \in T$, $x_{t,t} = 1$, and $x_{t,t'} = 0$ where $t' \in T$ and $t' \neq t$.

Lemma 2. $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} x_{t,t'} = 0$ if and only if \mathbf{x} is a labeling.

Proof. (\Rightarrow) If \mathbf{x} is a labeling, $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 = 0$ by Lemma 1, and also due to the second condition in Definition 2 we have $\sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} x_{t,t'} = 0$. Therefore,

$$\sum_{u \in V} \left(1 - \sum_{t \in T} x_{u,t}\right)^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} x_{t,t'} = 0.$$

(\Leftarrow) Now, suppose that $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} x_{t,t'} = 0$.

Since both terms are non-negative, to have sum of them equal to zero, they should be both zero. By Lemma 1, $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 = 0$ satisfies the first condition in Definition 2. Since $\sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} x_{t,t'} = 0$, it adds no penalty when for each $t \in T$, $x_{t,t} = 1$, and $x_{t,t'} = 0$ where $t' \in T$ and $t' \neq t$, which is the second condition in Definition 2. Therefore, \mathbf{x} is a labeling if $\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t})^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} x_{t,t'} = 0$. \square

Definition 3. Let $S_t \subset V$ be a subset of vertices labeled $t \in T$ by \mathbf{x} . \mathbf{x} is said to be feasible if and only if

(i) \mathbf{x} is a labeling;

(ii) The induced subgraph $G[S_t]$ contains t .

Lemma 3. The set of edges $E_c = \{\{u, v\} \mid x_{u,t} = x_{v,t'} = 1, \{u, v\} \in E, t, t' \in T, \text{ and } t \neq t'\}$ is a multi-way cut on G if \mathbf{x} is feasible.

Proof. If \mathbf{x} is feasible, it partitions G into k induced subgraphs, and if an induced subgraph is labeled by $t \in T$, then it contains t as well. Therefore, for each $\{u, v\} \in E$, if $x_{u,t} = x_{v,t'} = 1$ and $t \neq t'$ for $t, t' \in T$, then, u and v are not in the same induced subgraph. Hence, The set of edges E_c disconnects all the induced subgraphs, and accordingly all the terminals. By Definition 1, E_c is a multi-way cut. \square

Corollary 1. If \mathbf{x} is feasible, then the cost of a multi-way cut on G is computed as $|E_c|$ given in (3).

$$|E_c| = \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t' \neq t}} C(\{u, v\}) x_{u,t} x_{v,t'}. \quad (3)$$

Set $\alpha > \sum_{\{u,v\} \in E} C(\{u, v\})$. Recall the given QUBO model in (2):

$$H_{qubo}(\mathbf{x}) = \alpha \left(\sum_{u \in V} \left(1 - \sum_{t \in T} x_{u,t} \right)^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} x_{t,t'} \right) + \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t} x_{v,t'}.$$

Theorem 1. *Let $\mathbf{x}^* = \min_{\mathbf{x}} H_{qubo}(\mathbf{x})$. \mathbf{x}^* is feasible.*

Proof. To prove that \mathbf{x}^* is feasible, we need to consider the two conditions stated in Definition 3.

- (i) The second term adds a penalty as much as $\sum_{\{u,v\} \in E} C(\{u, v\})$ and α has been set greater than this penalty value. Therefore, for each $u \in V$ and $\{x_{u,t_1}^*, x_{u,t_2}^*, \dots, x_{u,t_k}^*\}$, we have $\sum_{t \in T} x_{u,t}^* = 1$, $x_{t,t}^* = 1$, and $x_{t,t'}^* = 0$ for $t, t' \in T$ and $t' \neq t$. \mathbf{x}^* is, therefore, a labeling by Definition 2,.
- (ii) Towards a contradiction, suppose that \mathbf{x}^* does not satisfy the second condition in Definition 3, and there is a subset of vertices $S_t \subset V$ labelled as $t \in T$ by which the induced subgraph $G[S_t]$ does not contain t . For a subset of vertices $S_{t'} \subset V$, let $G[S_{t'}]$ be a boundary induced subgraph labelled as $t' \in T$, which contains t' where $t' \neq t$. We set $\{u, v\} \in E$ as an edge that connects $G[S_t]$ and $G[S_{t'}]$ such that $u \in S_t$ and $v \in S_{t'}$. Since $t \neq t'$, the second term in $H_{qubo}(\mathbf{x}^*)$ adds penalties as much as $C(\{u, v\})$. In this case, there exists a \mathbf{x} that labels S_t by t' with fewer penalties (with no penalty for $\{u, v\}$ since u and v both have the same label by \mathbf{x}). In other words, $H_{qubo}(\mathbf{x}^*) > H_{qubo}(\mathbf{x})$, which is a contradiction.

Therefore, \mathbf{x}^* is feasible. □

Corollary 2. *The set of edges $E_m = \{\{u, v\} \mid x_{u,t}^* = x_{v,t'}^* = 1, \{u, v\} \in E, t, t' \in T, \text{ and } t \neq t'\}$ is a multi-way cut on G , and its cost shown by $|E_m|$ is defined as (4).*

$$|E_m| = \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t}^* x_{v,t'}^*. \quad (4)$$

Proof. It follows Lemma 3, Corollary 1, and Theorem 1. □

Theorem 2. *E_m is the minimum multi-way cut on G .*

Proof. Suppose that E_m is not the *minimum multi-way cut* on G , and $|E_m|$ is not the *minimum multi-way-cut cost*. In this case, there exists a multi-way cut E_c obtained from a feasible \mathbf{x} such that $|E_c| < |E_m|$. Therefore, by Corollary 1 and Corollary 2, we have

$$\sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t} x_{v,t'} < \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t}^* x_{v,t'}^*. \quad (5)$$

Since \mathbf{x} and \mathbf{x}^* are feasible, they both are labeling. By Lemma 2, (2), and (5), we have $H_{qubo}(\mathbf{x}) < H_{qubo}(\mathbf{x}^*)$, which is a contradiction. Thus, E_m is the *minimum multi-way cut* on G . \square

Theorem 3. $H_{qubo}(\mathbf{x}^*)$ is the minimum multi-way cost in G ; $H_{qubo}(\mathbf{x}^*) = |E_m|$.

Proof. By Theorem 1, \mathbf{x}^* is feasible and therefore \mathbf{x}^* is a labeling by Definition 3. By Lemma 2, $\alpha \left(\sum_{u \in V} (1 - \sum_{t \in T} x_{u,t}^*)^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} x_{t,t'}^* \right)$ is zero. Hence, we have

$$H_{qubo}(\mathbf{x}^*) = \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t}^* x_{v,t'}^*.$$

By Corollary 2, $H_{qubo}(\mathbf{x}^*) = |E_m|$. \square

3.2 Some examples

We now give a couple of concrete examples to illustrate our QUBO model.

Example 1. Figure 1a shows a simple undirected weighted graph $G(V, E, C)$ where $V = \{a, b, c, d, 1, 2\}$, and

$$E = \{\{a, b\}, \{b, d\}, \{d, c\}, \{a, c\}, \{a, 1\}, \{c, 2\}\}.$$

The set of terminals is $T = \{1, 2\}$ where $T \subset V$ and $k = 2$. Considering the given graph, $C(\{a, b\}) = 1$, $C(\{b, d\}) = 2$, $C(\{d, c\}) = 2$, $C(\{a, c\}) = 2$, $C(\{a, 1\}) = 5$, and $C(\{c, 2\}) = 4$. We set $\alpha = 25$ greater than $\sum_{\{u,v\} \in E} C(\{u, v\})$. Let $\mathbf{x} \in \{0, 1\}^{12}$ be a set of 12 binary variables such that

$$\mathbf{x} = \{x_{a,1}, x_{a,2}, x_{b,1}, x_{b,2}, x_{c,1}, x_{c,2}, x_{d,1}, x_{d,2}, x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}\}.$$

The QUBO model (2) for this graph can be formulated as follows.

$$\begin{aligned} H_{qubo}(\mathbf{x}) = & 25 \left(\sum_{u \in V} \left(1 - \sum_{t \in T} x_{u,t} \right)^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} x_{t,t'} \right) \\ & + \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t} x_{v,t'}. \end{aligned}$$

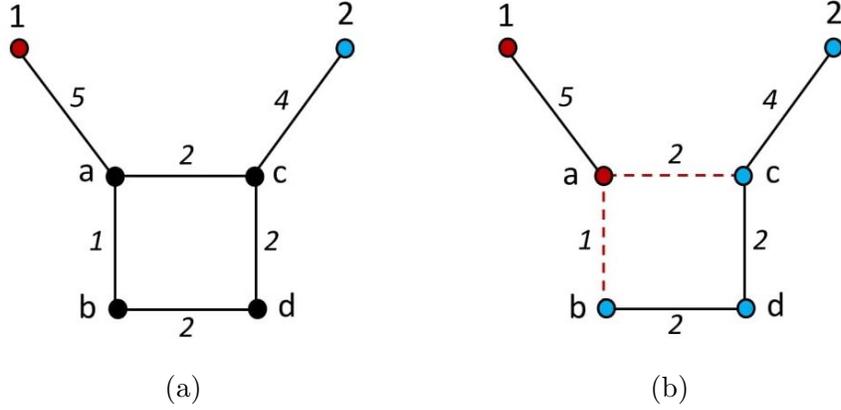


Figure 1: (a) A simple undirected weighted graph with two terminals. (b) The minimum multi-way-cut edges are shown by red dashed lines.

Let $H_1(\mathbf{x})$ and $H_2(\mathbf{x})$ be the first and second terms, respectively. We start with $H_1(\mathbf{x})$.

$$\begin{aligned}
H_1(\mathbf{x}) &= 25(-x_{a,1} - x_{a,2} + 2x_{a,1}x_{a,2} + 1) + 25(-x_{b,1} - x_{b,2} + 2x_{b,1}x_{b,2} + 1) \\
&+ 25(-x_{c,1} - x_{c,2} + 2x_{c,1}x_{c,2} + 1) + 25(-x_{d,1} - x_{d,2} + 2x_{d,1}x_{d,2} + 1) \\
&+ 25(-x_{1,1} - x_{1,2} + 2x_{1,1}x_{1,2} + 1) + 25(-x_{2,1} - x_{2,2} + 2x_{2,1}x_{2,2} + 1) \\
&+ 25x_{1,2} + 25x_{2,1}. \\
H_1(\mathbf{x}) &= -25x_{a,1} - 25x_{a,2} - 25x_{b,1} - 25x_{b,2} - 25x_{c,1} - 25x_{c,2} - 25x_{d,1} \\
&- 25x_{d,2} - 25x_{1,1} - 25x_{2,2} + 50x_{a,1}x_{a,2} + 50x_{b,1}x_{b,2} + 50x_{c,1}x_{c,2} \\
&+ 50x_{d,1}x_{d,2} + 50x_{1,1}x_{1,2} + 50x_{2,1}x_{2,2} + 150.
\end{aligned}$$

Next, we compute the second term.

$$\begin{aligned}
H_2(\mathbf{x}) &= x_{a,1}x_{b,2} + x_{a,2}x_{b,1} + 2x_{b,1}x_{d,2} + 2x_{b,2}x_{d,1} + 2x_{d,1}x_{c,2} \\
&+ 2x_{d,2}x_{c,1} + 2x_{a,1}x_{c,2} + 2x_{a,2}x_{c,1} + 5x_{a,1}x_{1,2} \\
&+ 5x_{a,2}x_{1,1} + 4x_{c,1}x_{2,2} + 4x_{c,2}x_{2,1}.
\end{aligned}$$

Adding both terms together, we have

$$\begin{aligned}
H_{qubo}(\mathbf{x}) &= H_1(\mathbf{x}) + H_2(\mathbf{x}) \\
&- 25x_{a,1} - 25x_{a,2} - 25x_{b,1} - 25x_{b,2} - 25x_{c,1} - 25x_{c,2} - 25x_{d,1} \\
&- 25x_{d,2} - 25x_{1,1} - 25x_{2,2} + 50x_{a,1}x_{a,2} + 50x_{b,1}x_{b,2} + 50x_{c,1}x_{c,2} \\
&+ 50x_{d,1}x_{d,2} + 50x_{1,1}x_{1,2} + 50x_{2,1}x_{2,2} + x_{a,1}x_{b,2} + x_{a,2}x_{b,1} \\
&+ 2x_{b,1}x_{d,2} + 2x_{b,2}x_{d,1} + 2x_{d,1}x_{c,2} + 2x_{d,2}x_{c,1} + 2x_{a,1}x_{c,2} \\
&+ 2x_{a,2}x_{c,1} + 5x_{a,1}x_{1,2} + 5x_{a,2}x_{1,1} + 4x_{c,1}x_{2,2} + 4x_{c,2}x_{2,1} \\
&+ 150.
\end{aligned}$$

By minimizing $H_{qubo}(\mathbf{x})$ over \mathbf{x} using an exact solver, we get the following results with $H_{qubo}(\mathbf{x}^*) = 3$: $x_{1,1}^* = 1$, $x_{1,2}^* = 0$, $x_{2,1}^* = 0$, $x_{2,2}^* = 1$, $x_{a,1}^* = 1$, $x_{a,2}^* = 0$, $x_{b,1}^* = 0$, $x_{b,2}^* = 1$, $x_{c,1}^* = 0$, $x_{c,2}^* = 1$, $x_{d,1}^* = 0$, $x_{d,2}^* = 1$.

Next, we create the minimum multi-way cut $E_m = \{\{u, v\} \mid x_{u,t}^* = x_{v,t'}^* = 1, \forall \{u, v\} \in E, \forall t, t' \in T, \text{ and } t \neq t'\}$ based on the results. Given $E = \{\{a, b\}, \{b, d\}, \{d, c\}, \{a, c\}, \{a, 1\}, \{c, 2\}\}$ as the set of edges, $E_m = \{\{a, b\}, \{a, c\}\}$. We also have $|E_m| = C(\{a, b\}) + C(\{a, c\}) = 2 + 1 = 3$, which is equal to $H_{qubo}(\mathbf{x}^*)$. Figure 1b illustrates the minimum multi-way cut for the given graph in Figure 1a.

Example 2. Figure 2a shows an undirected weighted graph $G(V, E, C)$ where $V = \{a, b, c, d, e, 1, 2, 3\}$, and $E = \{\{a, b\}, \{a, 1\}, \{a, 2\}, \{b, d\}, \{b, c\}, \{c, 3\}, \{d, 1\}, \{e, 1\}, \{1, 2\}, \{2, 3\}\}$. The set of terminals is $T = \{1, 2, 3\}$, where $T \subset V$ and $k = 3$. Considering the given graph, $C(\{a, b\}) = 2$, $C(\{a, 1\}) = 1$, $C(\{a, 2\}) = 5$, $C(\{b, d\}) = 2$, $C(\{b, c\}) = 1$, $C(\{c, 3\}) = 1$, $C(\{d, 1\}) = 2$, $C(\{e, 1\}) = 1$, $C(\{1, 2\}) = 1$, and $C(\{2, 3\}) = 2$. Let $\alpha = 20$ greater than the total sum of the weights. We need a set of 24 binary variables $\mathbf{x} \in \{0, 1\}^{24}$. The QUBO model (2) for this graph can be written as follows:

$$H_{qubo}(\mathbf{x}) = 20 \left(\sum_{u \in V} \left(1 - \sum_{t \in T} x_{u,t} \right)^2 + \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} x_{t,t'} \right) + \sum_{\{u,v\} \in E} \sum_{t \in T} \sum_{\substack{t' \in T \\ t \neq t'}} C(\{u, v\}) x_{u,t} x_{v,t'}.$$

By minimizing $H_{qubo}(\mathbf{x})$ over \mathbf{x} using an exact solver, we get 6 optimal solutions with the minimum possible energy. Let these solutions be \mathbf{x}_b^* , \mathbf{x}_c^* , \mathbf{x}_d^* , \mathbf{x}_e^* , \mathbf{x}_f^* and \mathbf{x}_g^* . We have $H_{qubo}(\mathbf{x}_b^*) = H_{qubo}(\mathbf{x}_c^*) = H_{qubo}(\mathbf{x}_d^*) = H_{qubo}(\mathbf{x}_e^*) = H_{qubo}(\mathbf{x}_f^*) = H_{qubo}(\mathbf{x}_g^*) = 7$. Based on the results shown in Table 1, we can define 6 minimum multi-way cuts as follows:

1. $E_m^b = \{\{a, 1\}, \{1, 2\}, \{a, b\}, \{2, 3\}, \{c, 3\}\}$,
2. $E_m^c = \{\{a, 1\}, \{1, 2\}, \{a, b\}, \{2, 3\}, \{b, c\}\}$,
3. $E_m^d = \{\{a, 1\}, \{1, 2\}, \{2, 3\}, \{b, c\}, \{b, d\}\}$,
4. $E_m^e = \{\{a, 1\}, \{1, 2\}, \{2, 3\}, \{c, 3\}, \{b, d\}\}$,
5. $E_m^f = \{\{a, 1\}, \{1, 2\}, \{2, 3\}, \{b, c\}, \{d, 1\}\}$,
6. $E_m^g = \{\{a, 1\}, \{1, 2\}, \{2, 3\}, \{c, 3\}, \{d, 1\}\}$.

Figure 2 shows the corresponding minimum multi-way cuts for the graph given in Figure 2a. The total cost of all the minimum multi-way cuts is 7.

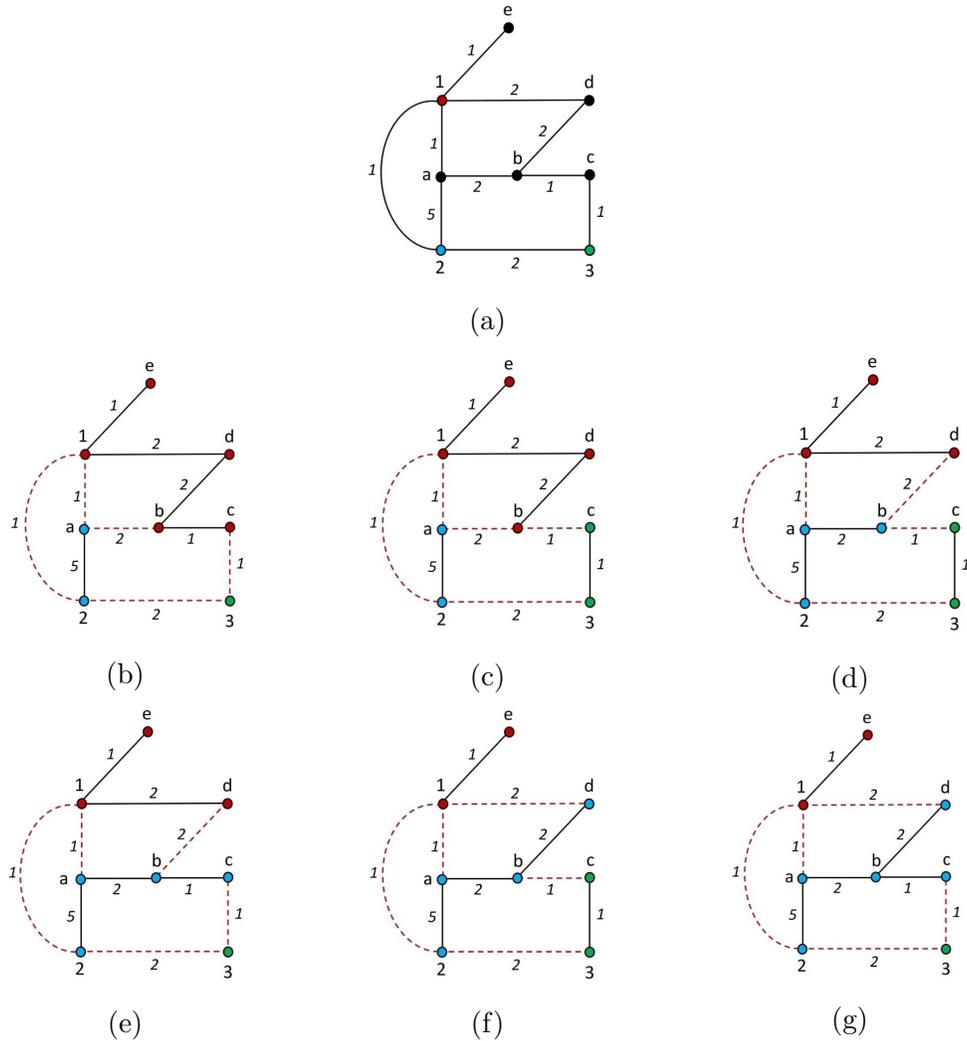


Figure 2: (a) A simple undirected weighted graph with three terminals. (b-g) The minimum multi-way cuts obtained from \mathbf{x}_b^* , \mathbf{x}_c^* , \mathbf{x}_d^* , \mathbf{x}_e^* , \mathbf{x}_f^* , and \mathbf{x}_g^* , respectively. The minimum multi-way-cut edges are shown by red dashed lines.

4 An application of the minimum multi-way cut in Computer Vision

Computer Vision is aiming at inferring high-level perception about the real world from digital images in a way similar to the human vision system for tasks such as image content and context description including image filtering (e.g. noise removal), image segmentation, objects detection and classification, and 3D perception tasks such as depth estimation and or 3D scene reconstruction from a single or multiple images. Both the human vision system and computer vision tend to perform well on different subset of the above tasks. For example, computers still struggle to interpret and perceive the real world through images although humans perform this task effortlessly. Early vision problems often use perception model based on a labeling paradigm where a set of image features is described using distinct markers

Table 1: The optimal solutions of minimizing $H_{qubo}(\mathbf{x})$ in Example 2.

$u \in V$	\mathbf{x}^*	\mathbf{x}_b^*	\mathbf{x}_c^*	\mathbf{x}_d^*	\mathbf{x}_e^*	\mathbf{x}_f^*	\mathbf{x}_g^*
1	$x_{1,1}^*$	1	1	1	1	1	1
	$x_{1,2}^*$	0	0	0	0	0	0
	$x_{1,3}^*$	0	0	0	0	0	0
2	$x_{2,1}^*$	0	0	0	0	0	0
	$x_{2,2}^*$	1	1	1	1	1	1
	$x_{2,3}^*$	0	0	0	0	0	0
3	$x_{3,1}^*$	0	0	0	0	0	0
	$x_{3,2}^*$	0	0	0	0	0	0
	$x_{3,3}^*$	1	1	1	1	1	1
a	$x_{a,1}^*$	0	0	0	0	0	0
	$x_{a,2}^*$	1	1	1	1	1	1
	$x_{a,3}^*$	0	0	0	0	0	0
b	$x_{b,1}^*$	1	1	0	0	0	0
	$x_{b,2}^*$	0	0	1	1	1	1
	$x_{b,3}^*$	0	0	0	0	0	0
c	$x_{c,1}^*$	1	0	0	0	0	0
	$x_{c,2}^*$	0	0	0	1	0	1
	$x_{c,3}^*$	0	1	1	0	1	0
d	$x_{d,1}^*$	1	1	1	1	0	0
	$x_{d,2}^*$	0	0	0	0	1	1
	$x_{d,3}^*$	0	0	0	0	0	0
e	$x_{e,1}^*$	1	1	1	1	1	1
	$x_{e,2}^*$	0	0	0	0	0	0
	$x_{e,3}^*$	0	0	0	0	0	0
Figure	2	2b	2c	2d	2e	2f	2g

also known as labels. Generally, a labeling problem in Computer Vision is defined over a set of image features which consider local variations of pixel intensities around a given location. Such concepts have allowed to mathematically define (real world) edges or segments in images. Labeling problems can be resolved through an energy minimization scheme based on an energy function derived from probabilistic graphical models such as Markov Random Field. For a detailed explanation of how this energy function is defined, we refer the interested reader to [17, 18, 26]). This energy function is usually composed of terms: the first term penalizes the solutions when they are inconsistent with the observed data, and the second one is a regularisation term that imposes some constraints for the spatial inconsistencies [23]. In the following section, we first represent a pixel-labeling problem namely image restoration by an energy function, and then explain how it can be minimized by solving the *minimum multi-way cut* problem on a certain graph.

4.1 Image restoration

Image restoration is a family of inverse problems to recover an original high-quality image from a corrupted input image (see Figure 3). There are some reasons that corruption may occur such as the image capture process (e.g., noise, lens blur), post-processing (e.g., JPEG compression), or photography in non-ideal conditions (e.g., haze, motion blur) [12]. Image restoration can be modeled by a labeling problem where a set of pixels is labeled by some quantities. Since image restoration consists in recovering as much as possible the image original pixel intensities, the set of labels should contain the original intensities. We will

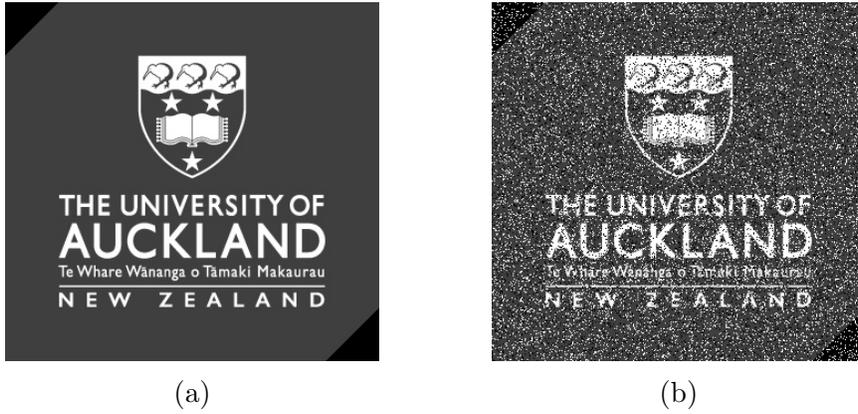


Figure 3: (a) The original image, (b) The noisy image.

introduce the problem notations before defining an image restoration energy function.

In the most general form, a digital image is a function $I : P \rightarrow H$ where $P = \{(i, j) \mid 0 \leq i < n, 0 \leq j < m\}$ is a set of two-dimensional spatial coordinates, $H = \{0, \dots, h - 1\}$ is a set of signal values, and $n, m, h \in \mathbb{N}$. The coordinate $(i, j) \in P$ is referred to as a pixel, and $I(i, j)$ is called intensity of the image at pixel location (i, j) . In fact, a digital image consists of n lines of m pixels with intensities varying between 0 and $h - 1$ (typically, gray-scale image intensities are encoded as 8-bit integers and up to 16-bit integers for medical applications). The pixels in P are related to each other based on a neighborhood system. Let N defined in (6) be a 4-neighborhood system by which each pixel has at most four neighboring pixels.

$$N = \{ \{(i, j), (i', j')\} \mid (i, j) \in P, (i', j') \in \{(i \pm 1, j), (i, j \pm 1)\}, 0 \leq i' < n, 0 \leq j' < m \}. \quad (6)$$

Let's define $L = \{0, \dots, h - 1\}$ as the set of labels. The main goal here is to label each pixel in P with a value in L . Therefore, we define \mathbf{w} as a vector of integer variables such that $\mathbf{w} = (w_{i,j})_{(i,j) \in P}$ where $w_{i,j} \in L$. Given I as the input noisy image, image restoration can be represented by the energy function $F : L^{n \times m} \rightarrow \mathbb{R}$ as follows [3].

$$F(\mathbf{w}) = \sum_{(i,j) \in P} (I(i, j) - w_{i,j})^2 + \lambda \sum_{\{(i,j), (i',j')\} \in N} \delta(w_{i,j}, w_{i',j'}), \quad (7)$$

$$\delta(w_{i,j}, w_{i',j'}) = \begin{cases} 0, & \text{if } w_{i,j} = w_{i',j'}; \\ 1, & \text{otherwise,} \end{cases}$$

where λ is a positive integer, and $I(i, j)$ is the observed intensity of pixel (i, j) on the noisy image I . The first term is to compute the cost of choosing the label $w_{i,j}$ for the pixel (i, j) , and the second term is for the contextual constraint which encodes a preference about the labels of the neighboring pixels. Here, δ insures that the intensities of a neighborhood of

pixels present some coherence and generally do not change abruptly. In 2001, Boykov et al. [3] showed that the problem of minimizing such energy function, which is known as Potts Model, can be solved by finding the *minimum multi-way cut* on a certain graph (Theorem 7.1 [3, p.1231]).

Algorithm 1 (Input: P , N and L ; Output: $G(V, E, C)$)

```

1: for each pixel  $(i, j) \in P$  do  $\triangleright 0 \leq i < n, 0 \leq j < m$ 
2:    $V \leftarrow V \cup \{(i, j)\}$ 
3: for each  $\{(i, j), (i', j')\} \in N$  do
4:    $E \leftarrow E \cup \{(i, j), (i', j')\}$ 
5:    $C(\{(i, j), (i', j')\}) = \lambda$ 
6: for  $l \in L$  do
7:    $V \leftarrow V \cup \{l\}$ 
8: for  $(i, j) \in V$  do
9:   for  $l \in L$  do
10:     $E \leftarrow E \cup \{l, (i, j)\}$ 
11:     $C(\{l, (i, j)\}) = K_{i,j} - (I(i, j) - l)^2$ 
12:  $\triangleright K_{i,j}$  is a constant greater than  $(I(i, j) - l)^2$  for all  $l \in L$ 

```

Algorithm 1 shows how to create such graph for image restoration tasks. Let's define an image $I : \{0, \dots, 3\}, \{0, \dots, 3\} \rightarrow \{0, 1, 2\}$ as an example to show how Algorithm 1 works. We first define L , P , and N as follows.

$$\begin{aligned}
L &= \{0, 1, 2\}. \\
P &= \{(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (1, 2), \\
&\quad (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}. \\
N &= \{\{(0, 0), (0, 1)\}, \{(0, 0), (1, 0)\}, \{(0, 1), (0, 2)\}, \{(0, 1), (1, 1)\}, \\
&\quad \{(0, 2), (0, 3)\}, \{(0, 2), (1, 2)\}, \{(0, 3), (1, 3)\}, \{(1, 0), (1, 1)\}, \\
&\quad \{(1, 1), (2, 0)\}, \{(1, 1), (1, 2)\}, \{(1, 1), (2, 1)\}, \{(1, 2), (1, 3)\}, \\
&\quad \{(1, 2), (2, 2)\}, \{(1, 3), (2, 3)\}, \{(2, 0), (2, 1)\}, \{(2, 0), (3, 0)\}, \\
&\quad \{(2, 1), (2, 2)\}, \{(2, 1), (3, 1)\}, \{(2, 2), (2, 3)\}, \{(2, 2), (3, 2)\}, \\
&\quad \{(2, 3), (3, 3)\}, \{(3, 0), (3, 1)\}, \{(3, 1), (3, 2)\}, \{(3, 2), (3, 0)\}\}.
\end{aligned}$$

Next, we construct the discussed graph based on Algorithm 1. Figure 4a shows this graph. Let E_m be the *minimum multi-way cut* on this graph. Based on the weight initialization in Algorithm 1, in $G(V, E/E_m)$ each pixel vertex is connected to exactly one label vertex, which can be interpreted as the associated label to the corresponding pixel [3]. Figure 4b provides an illustration to show how the *minimum multi-way cut* for the structured graph in Figure 4a would determine the associated labels to each pixel.

In the next section, we show some experimental results to restore different noisy images using the QUBO model (2) based on a D-Wave hybrid quantum-classical solver.

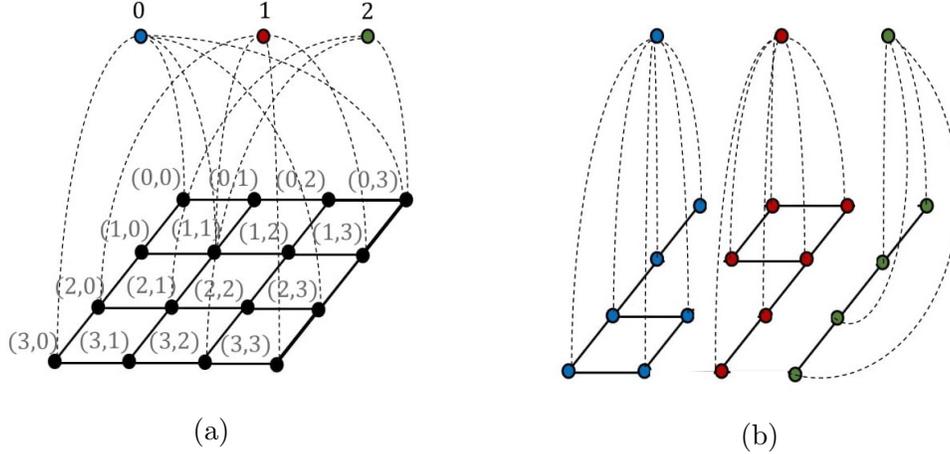


Figure 4: (a) The graph structure for a noisy image with $n = 4$, $m = 4$ and a set of labels as $L = \{0, 1, 2\}$. Each vertices "0", "1" and "2" are connected to each pixel $(i, j) \in P$. To simplify the illustration, we show only a small number of these edges. The edges between pixels have been added based on N . (b) An illustration for the *minimum multi-way cut* by which each pixel could be labeled uniquely.

4.2 Image restoration based on a D-Wave hybrid solver

Due to the scarcity of physical qubits on D-Wave QPUs, in some cases, input data are too large to fit onto the QPU and be solved directly by Quantum Annealing. For this reason, D-Wave company has prepared different hybrid solvers to overcome such size limitations by combing classical and quantum approaches for problem-solving. These solvers run multiple solvers in parallel and return the best solution from a pool of results. In our experiment, we used a D-Wave hybrid solver for image restoration based on finding the *minimum multi-way cut* on a certain graph as discussed earlier. We created an image with size 100×100 pixels with 10 colors where $n = m = 100$ and $h = 10$. Therefore, we needed a set of 10 labels as $L = \{0, \dots, 9\}$. We used *salt-and-pepper* noise with different intensities to make various levels of noise on the original image. Sharp and sudden disturbances in the image signal can make such noise, and it randomly changes the original intensities to white and black pixels. Figure 5 shows the original image and the noisy images. The noise percentage has been shown by p_n (eg., when $p_n = 20\%$, it means 20 percent of the original pixel intensities have been randomly changed into white and black pixels to simulate the *salt-and-pepper* noise). Next, for each input noisy image, we created an undirected weighted graph based on Algorithm 1, and then formulated the QUBO model (2) to find the *minimum multi-way cut* on the graph. After minimizing the QUBO model by the D-Wave hybrid solver, we changed noisy pixel intensities to the obtained labels to restore the original pixels. Figure 5 shows the image restoration results for the corresponding noisy images. To compare the original intensities with the restored ones, we define *root-mean-squared (rms)* defined in (8) which

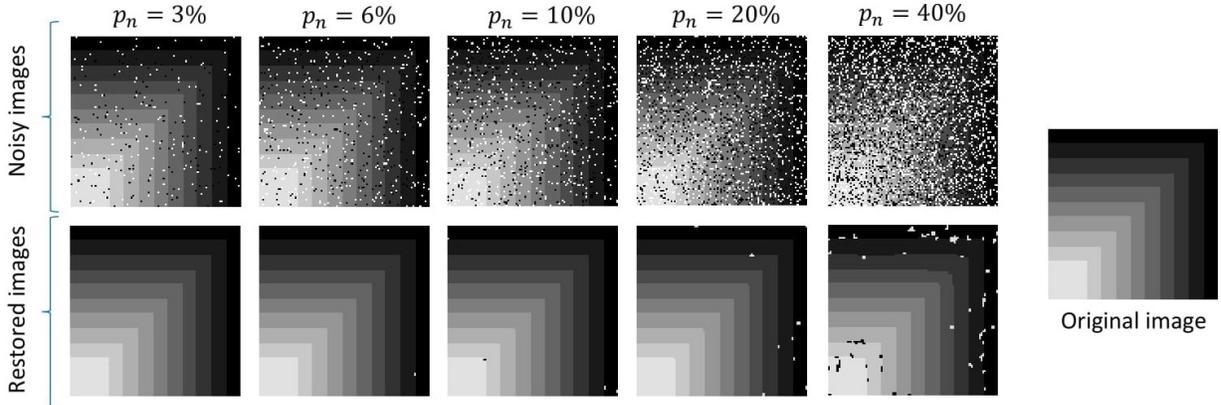


Figure 5: Image restoration results from the D-Wave hybrid solver. The first row shows the defined noisy images with different levels of noise, and the second row represents the corresponding restored images.

Table 2: *rms* results for comparing the original intensities with the restored intensities

Noisy images	$p_n = 3\%$	$p_n = 6\%$	$p_n = 10\%$	$p_n = 20\%$	$p_n = 40\%$
<i>rms</i>	0	4.04	4.53	12.75	23.44

shows the percentage of intensity variation (bad restored intensities).

$$rms = \sqrt{\frac{1}{nm} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (I_O(i, j) - I_R(i, j))^2}, \quad (8)$$

where I_O is the original image before applying the noise, I_R is the restored image, and nm is the total number of pixels. Table 2 shows the *rms* results for our benchmark.

5 Conclusion

We have provided an efficient QUBO formulation for the *minimum multi-way cut* problem that requires only $O(k|V|)$ logical binary variables, where k is the number of terminals that need to be edge-separated in a graph of order $|V|$. Our problem has applications to Computer Vision where the input sizes are quite large so minimizing the problem size for a quantum annealing computer is of importance. To model one of these applications, we showed how a noisy image can be restored based on the *minimum multi-way cut* problem and our QUBO model. We defined an image with 10 color intensities and applied *salt & pepper* noise to it. Our experimental implementation on a D-Wave hybrid quantum-classical solver resulted in acceptable restored images with respect to the amount of applied noise. In the future, we hope to find good logical-to-physical embeddings of our QUBOs onto existing quantum hardware (e.g. D-Wave Chimera, Pegasus, or Zephyr architectures). We also want to experiment further with hybrid-quantum computations using our QUBO formulation and compare them against state-of-the-art classical algorithms.

Acknowledgments

We thank Cristian Calude and Richard Hua for their helpful discussions on this paper.

References

- [1] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.
- [2] A. Avidor and M. Langberg. The multi-multiway cut problem. *Theoretical Computer Science*, 377(1-3):35–42, 2007.
- [3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.
- [4] G. Călinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 48–52, 1998.
- [5] C. S. Calude and M. J. Dinneen. Solving the broadcast time problem using a D-Wave quantum computer. In *Advances in Unconventional Computing*, pages 439–453. Springer, 2017.
- [6] C. S. Calude, M. J. Dinneen, and R. Hua. QUBO formulations for the graph isomorphism problem and related problems. *Theoretical Computer Science*, 701:54–69, 2017.
- [7] M.-C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximal integer multiflow: a survey. *European Journal of Operational Research*, 162(1):55–69, 2005.
- [8] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.
- [9] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven. What is the computational value of finite-range tunneling? *Physical Review X*, 6(3):031015, 2016.
- [10] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [11] Y. Dinitz. Dinitz algorithm: The original version and Even’s version. In *Theoretical computer science*, pages 218–240. Springer, 2006.

- [12] N. Elron, S. S. Yuval, D. Rudoy, and N. Levy. Blind image restoration without prior knowledge. *arXiv preprint arXiv:2003.01764*, 2020.
- [13] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.
- [14] L. Ford Jr and D. Fulkerson. Flows in networks (1962) princeton univ.
- [15] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.
- [16] J. King, S. Yarkoni, J. Raymond, I. Ozfidan, A. D. King, M. M. Nevisi, J. P. Hilton, and C. C. McGeoch. Quantum annealing amid local ruggedness and global frustration. *Journal of the Physical Society of Japan*, 88(6):061007, 2019.
- [17] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [18] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, Berlin, Heidelberg, 1995.
- [19] A. Lucas. Ising formulations of many NP problems. *Frontiers in physics*, 2:5, 2014.
- [20] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 11–20, 2008.
- [21] C. C. McGeoch. Adiabatic quantum computation and quantum annealing: Theory and practice. *Synthesis Lectures on Quantum Computing*, 5(2):1–93, 2014.
- [22] J. Naor and L. Zosin. A 2-approximation algorithm for the directed multiway cut problem. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 548–553. IEEE, 1997.
- [23] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):1068–1080, 2008.
- [24] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. Cornell University, 1999.
- [25] T. Verma and D. Batra. Maxflow revisited: An empirical comparison of maxflow algorithms for dense vision problems. In *BMVC*, pages 1–12, 2012.
- [26] C. Wang, N. Komodakis, and N. Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

- [27] M. Xiao. Simple and improved parameterized algorithms for multiterminal cuts. *Theory of Computing Systems*, 46(4):723–736, 2010.
- [28] R. Yaacoby, N. Schaar, L. Kellerhals, O. Raz, D. Hermelin, and R. Pugatch. A comparison between d-wave and a classical approximation algorithm and a heuristic for computing the ground state of an Ising spin glass. *arXiv preprint arXiv:2105.00537*, 2021.
- [29] W.-C. Yeh. A simple algorithm for the planar multiway cut problem. *Journal of Algorithms*, 39(1):68–77, 2001.