

Article

# A QUBO Formulation of the Stereo Matching Problem for D-Wave Quantum Annealers

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

<sup>1</sup> CU-UAEM Valle de Chalco, Hermenegildo Galeana 3, Valle de Chalco 56615, Estado de México, Mexico; wdelacruz@uaemex.mx

<sup>2</sup> Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias. Ave., Eugenio Garza Sada 2501, Monterrey 64849, NL, Mexico

<sup>3</sup> US Naval Research Laboratory, 4555 Overlook Ave., SW Washington, DC 20375, USA; marco.lanzagorta@nrl.navy.mil

\* Correspondence: svenegas@itesm.mx or salvador.venegas-andraca@keble.oxon.org; Tel.: +52-55-5864-5555

Received: 6 August 2018; Accepted: 8 October 2018; Published: 12 October 2018



**Abstract:** In this paper, we propose a methodology to solve the stereo matching problem through quantum annealing optimization. Our proposal takes advantage of the existing Min-Cut/Max-Flow network formulation of computer vision problems. Based on this network formulation, we construct a quadratic pseudo-Boolean function and then optimize it through the use of the D-Wave quantum annealing technology. Experimental validation using two kinds of stereo pair of images, random dot stereograms and gray-scale, shows that our methodology is effective.

**Keywords:** quantum annealing; stereo matching; quantum algorithms

## 1. Introduction

Computer vision is an interdisciplinary field of research with almost six decades of theoretical and algorithmic developments [1,2] that focuses on developing mathematical techniques and algorithms that aim at enabling computers to identify, analyze, and understand information from elements of imagery [3]. Computer vision has many links and common interests with Artificial Intelligence and Machine Learning.

Stereo vision refers to the ability to extract information about the 3-D structure and distance of a scene from two or more images taken from different viewpoints. The most basic stereo system consists of two cameras (left and right) and any stereo system must solve two problems:

1. The stereo matching problem [4]: Which parts of the left and right images are projections of the same scene element?
2. The reconstruction problem, which is stated as follows: given a number of corresponding parts of the left and right images, what can we say about the 3-D locations and structures of the observed objects?

In this paper, we focus on the stereo matching problem using a basic stereo system. This problem is difficult to solve because some parts of the scene are visible only by either the left or right camera but not by both; therefore, a stereo system must also be able to select the image parts to be matched [5]. Stereo vision has many applications, such as in photogrammetry [6], stereo-based head tracking [7], volumetric and 3-D surface reconstruction [8], video-based walkthroughs [9], and stereo-based autonomous navigation [10], among others.

Nowadays, there is an increasing amount of interest in applying quantum computing techniques on machine learning tasks. The research area of quantum machine learning (QML) [11,12] has as its

main objective designing quantum algorithms and data representation using quantum states that outperform classical algorithms in machine learning problems. Examples of recent progress in QML are the proposals of quantum principal component analysis of classical data [13] and quantum support vector machine [14] (both papers use quantum states that encode classical data as input), as well as deep quantum learning [15] and sampling of a quantum Boltzman machine [16] (these two papers have classical data as input). More recent advances in QML can be found in [17,18].

Quantum hardware technologies such as the quantum circuit-based IBMQ computer [19] and the D-Wave [20] quantum annealer machine have motivated the development of applications in QML. In particular, the D-Wave computer has evolved and grown significantly with respect to the number of qubits, allowing the posing of problems of interest [21–24]. For instance, the current generation of D-Wave 2000Q processors has almost 2048 working superconducting qubits connected in a graph topology known as Chimera.

In this paper, we present a methodology for writing a Quadratic Unconstrained Binary Optimization (QUBO) formulation of the stereo matching problem via a graph-cut approach, followed by using this QUBO formulation to simulate a quantum annealing algorithm on D-Wave's simulation software. This paper is meant to be a contribution to the nascent area of quantum algorithms to solve relevant problems in science, engineering and other fields, like finance.

Our method could also be used to implement NP-hard problem instances in novel architectures of classical hardware, like the Ising model architectures proposed by Fujitsu [25] and Hitachi [26] research laboratories (examples of NP-hard problems are combinatorial optimization problems. A concise introduction to NP-hardness can be found in [27]). Furthermore, since the stereo matching problem is of paramount importance in computer vision, several classical algorithms have been developed to solve it, among them, graph-cut-based methods (we provide a brief account of algorithms for stereo matching in Section 2). Although the stereo matching problem is in the P class, a minimum upper bound for the computational complexity is unknown; hence, the development of novel approaches is pertinent. In this sense, the proposal presented in this manuscript combined with present and future research on quantum annealing-based algorithms could be used in future work to develop new and less expensive algorithms.

Our work consists of two main goals: (a) to express a problem of interest as a QUBO problem and (b) to find a minor embedding of the QUBO problem into the Chimera graph architecture. Before applying our methodology, we represent the stereo matching problem as a graph-cut problem by constructing an edge weighted graph with two special vertices. This weighted graph has the following property: finding a subset of edges with minimum cost whose removal disconnects the special vertices is equivalent to solving the stereo matching problem. Therefore, for step (a), we construct a pseudo-Boolean expression of degree three for the graph-cut problem, and we use a reduction method to obtain a QUBO expression to be minimized. For step (b), a minor embedding is found by using the software tools provided by the D-Wave solver application programming interface (SAPI). To validate our methodology for the solution to the stereo matching problem, we present the results of the minimization of the QUBO expression using the classic solver `qbsolv` developed by D-Wave.

This paper is organized as follows: In Section 2, we succinctly introduce the basic notions of optimization via quantum annealing and we describe our methodology for solving the stereo matching problem using QUBO and D-Wave technology; in Section 3, we present our experiments and results; finally, we present our conclusions in Section 4.

## 2. Results

### 2.1. Quantum Annealing

Quantum annealing (QA) [28–33] can be seen as a quantum version of Classical Simulated Annealing (SA) [32–35]. SA uses the process of annealing in which a solid in a thermal bath is heated up by increasing the temperature, followed by a cooling by slowly lowering the temperature of the

bath. The annealing allows the particles in the solid to go from a random configuration or high energy to a lattice configuration or low energy. This process was found to be equivalent to the problem of finding the configuration with the minimum amount of energy or the cost function of a given combinatorial optimization problem. QA employs quantum fluctuations to anneal the system down to its minimum energy state [33]. The most used physical system for QA is the Ising spin glass model on  $N$  particles described by

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

where  $\sigma_i^z$  is the Pauli matrix  $z$  acting on particle  $i$ ,  $h_i$  describes the magnetic field on particle  $i$ , and  $J_{ij}$  is the coupling strength between particles  $i$  and  $j$ . The minimum energy state or ground state of the Hamiltonian  $H_p$  corresponds to a configuration  $\mathbf{s} = (s_1, \dots, s_N) \in \{+1, -1\}^N$  of spins that minimizes the following energy function

$$E(\mathbf{s}) = \sum_{1 \leq i \leq N} h_i s_i + \sum_{1 \leq i < j \leq N} J_{ij} s_i s_j. \quad (2)$$

It is known that the problem of finding a configuration  $\mathbf{s}^*$  with minimum energy  $E(\mathbf{s}^*)$  is an NP-complete problem [36] (although some special cases of the Ising model can be solved in polynomial time. However, those cases are not part of our study). A related problem with the same complexity is obtained by changing the variable  $x_i = (1 + s_i)/2$  for  $i = 1, \dots, N$ . In other words, finding a configuration  $\mathbf{x} \in \{0, 1\}^N$  such as  $E(\mathbf{x})$  that is minimum also corresponds to an NP – complete problem [37].

One of the schemes to realize QA is through adiabatic quantum evolution from the ground state of an initial Hamiltonian to the ground state of a final Hamiltonian which corresponds to the solution of a given problem [38]. According to this scheme, a time-dependent Hamiltonian takes the form

$$H(\tau) = A(\tau)H_0 + B(\tau)H_p \quad (3)$$

where  $\tau = t/t_a$  for  $0 \leq t \leq t_a$ , is  $t_a$  is the total annealing time, and the initial Hamiltonian  $H_0 = -\sum_{1 \leq i \leq N} \sigma_i^x$  is responsible for quantum tunnelling among the localized classical states corresponding to the eigenstates of the Hamiltonian  $H_p$ . In the current generation of the D-Wave 2000Q processor, functions  $A(\tau)$  and  $B(\tau)$  are defined so that at time  $\tau = 0$ , the influence of the Hamiltonian  $H_0$  is predominant against  $H_p$ . As time evolves from  $\tau = 0$  to  $\tau = 1$ , the influence of the Hamiltonian  $H_p$  increases, while  $H_0$  fades away.

One then can solve the time-dependent Schrödinger equation with the Hamiltonian  $H(t)$  to obtain an approximate solution to the dynamics of the system. 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  $\lambda_l(\tau)$ . Then,

$$\tilde{H}(\tau)|l; \tau\rangle = \lambda_l(\tau)|l; \tau\rangle \quad (4)$$

with

$$\lambda_0(\tau) \leq \lambda_1(\tau) \leq \dots \leq \lambda_{2^N-1}(\tau). \quad (5)$$

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

$$\lim_{t_a \rightarrow \infty} |\langle l = 0; \tau = 1 | \psi(t_a) \rangle| = 1 \quad (6)$$

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

$$t_a \gg \frac{\epsilon_{max}}{g_{min}^2} \tag{7}$$

where

$$g_{min} = \min_{0 \leq \tau \leq 1} (\lambda_1(\tau) - \lambda_0(\tau)) \tag{8}$$

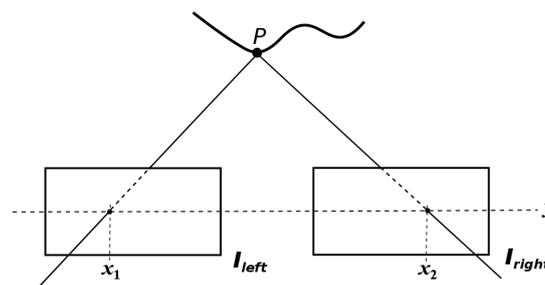
and

$$\epsilon_{max} = \max_{0 \leq \tau \leq 1} |\langle l = 1; \tau | \frac{d\tilde{H}}{d\tau} | l = 0; \tau \rangle|. \tag{9}$$

A more rigorous formulation of the Adiabatic theorem can be found in [39]. It must be noted that in realistic physical settings, including the D-Wave processor, the QA dynamics of a system do not necessary evolve adiabatically. There are several reasons including thermal fluctuations and additive noise from the environment. Other considerations to take into account when solving problems on a D-Wave processor are the size of the device or number of qubits in the current architecture and the precision of the Ising parameters, such as the magnetic field values  $h_i$  and the coupler strength  $J_{ij}$  between pairs of qubits. A concise introduction to quantum annealing algorithms, their potential relevance for the advancement of key problems in theoretical computer science, as well as mathematical methods to implement quantum annealing algorithms in D-Wave quantum annealers can be found in [27].

### 2.1.1. QUBO Formulation Approach to the Stereo Matching Problem

The stereo matching problem is of key importance in computer vision [5,40]. Without any loss of generality, a stereo setup consists of two spatially separated cameras located at the same height with parallel optical axes. A stereo pair of images,  $I_l$  and  $I_r$ , are the recorded images by the left and right cameras,  $c_l$  and  $c_r$ , respectively. For a point  $P$  in a scene observed by a stereo setup, it (point  $P$ ) will be projected into the left and right images with intensities  $I_l(x_1, y)$  and  $I_r(x_2, y)$ , respectively. Notice that, in a stereo setup, a point  $P$  will be projected at the same row  $y$  in both left and right images, whilst determining the columns  $x_1$  and  $x_2$  of projection is a difficult problem [4]. Let us now state the following problem.



**Figure 1.** Example of the projection of a point  $P$  on the image planes of the left and right images. As can be seen in this illustration, the point  $P$  is projected to the positions  $x_1$  and  $x_2$  of the left and right images, respectively. The difference  $x_2 - x_1$  is called the disparity of point  $P$ .

**Problem 1** (Stereo matching). *Given a stereo pair of images  $(I_l, I_r)$ , find, for every pixel  $I_l(x_1, y)$  in the left image, its corresponding projection  $I_r(x_2, y)$  in the right image (see Figure 1).*

The solution to problem 1 involves calculating the disparity for every pixel in the left image. If the calibration parameters of the stereo setup are given [40], then it is possible to infer the depth at every point in the scene. Early approaches to solve problem 1 are those based on block matching techniques [4], dynamic programming [41], belief propagation [42] and graph-cut [43] based techniques, among others. In the following section, we formulate the stereo matching problems in terms of graph-cuts.

### 2.1.2. Energy Function

Let  $\mathcal{P}$  be the image domain or set of pixels in the left image and let  $\mathcal{L}$  be the set of labels or disparities; a labeling is a map

$$l : \mathcal{P} \rightarrow \mathcal{L} \quad (10)$$

such that for each pixel  $p \in \mathcal{P}$ ,  $l$  assigns a label  $l(p) = l_p$  to the pixel  $p$ .

Given a labeling  $l$ , its cost is defined as

$$\mathcal{E}(l) = \sum_{p \in \mathcal{P}} D_p(l_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(l_p, l_q) \quad (11)$$

where  $D_p$  models the cost of assigning label  $l_p$  to the pixel  $p$ ,  $V_{pq}$  models the cost of assigning  $l_p$  to  $p$  and  $l_q$  to  $q$ , with  $p$  and  $q$  being adjacent pixels, and  $\mathcal{N}$  is the set of neighbor pairs  $\{p, q\}$ .

Graph-cut techniques consist of constructing a graph with weighted edges and two special vertices,  $s$  and  $t$ . In this model, finding a cut with minimum cost that separates  $s$  and  $t$  is equivalent to minimizing function  $\mathcal{E}(l)$  [44,45].

### 2.1.3. Graph Construction

The following graph construction is based on the Boykov and Kolmogorov approach for Min-Cut/Max-Flow Algorithms for Energy Minimization [44].

Let  $G = (V, E, c)$  be an edge weighted graph where  $V$  is the set of vertices with two special vertices, the source  $s$  and the sink  $t$ ,  $E$  is the set of undirected edges, and  $c : E \rightarrow \mathbb{Z}^+$  is a function that assigns a positive value to each edge in  $E$ . A cut in  $G$  is a set of edges whose removal from  $E$  separates the source from the sink, and its cost is the sum of their weights. Let  $\mathcal{L} = \{0, \dots, L\}$  be the set of possible disparities or labels. The graph  $G$  is constructed as follows:

1. To each pixel  $p$  in the image, we associate a chain composed of  $L + 2$  vertices, say  $p_0, p_1, \dots, p_{L+1}$ . These vertices are connected by edges called  $t$ -links,  $\{e_{p_0, p_1}, e_{p_1, p_2}, \dots, e_{p_L, p_{L+1}}\}$ , where  $e_{p_0, p_1} = \{p_0, p_1\}$ ,  $e_{p_1, p_2} = \{p_1, p_2\}$ ,  $\dots$ ,  $e_{p_L, p_{L+1}} = \{p_L, p_{L+1}\}$ . For a disparity range from 0 to  $L$  values, and for each pixel  $p$ , we will have  $L + 2$  vertices in the chain and  $L + 1$   $t$ -links between these vertices; each edge defines a labeling with one specific value. Additionally, there are two  $t$ -links that connect the first vertex with the source and the last vertex with the sink. Hence, there exist  $L + 3$  such edges in the graph for each pixel  $p$ . The weights of those edges directly depend on the function  $D_p$ , which specifies the cost of applying a specific label to a pixel.
2. To each pair of neighbor pixels,  $p$  and  $q$ , there will be links that connect the corresponding chains with edges called  $n$ -links. For instance, let  $p_0, p_1, \dots, p_{L+1}$  and  $q_0, q_1, \dots, q_{L+1}$  be the chain vertices for two neighbor pixels  $p$  and  $q$ ; the  $n$ -links between chains are  $e_{p_0, q_0} = \{p_0, q_0\}$ ,  $e_{p_1, q_1} = \{p_1, q_1\}$ ,  $\dots$ ,  $e_{p_{L+1}, q_{L+1}} = \{p_{L+1}, q_{L+1}\}$ . Usually a 4-neighborhood is assumed for every pixel. The weights of these edges should reflect the penalty when assigning different labels to neighboring pixels.

For a given stereo pair of images of size  $N \times M$  pixels and a disparity range  $L$ , the total number of vertices in the constructed graph is  $(L + 2)NM + 2$  and the number of edges is  $NM(L + 3) + ((M - 1)N + (N - 1)M)(L + 2)$ . In Figure 2, we present an example of the graph topology for an image of  $5 \times 5$  pixels and  $L = 3$ , in which we show the  $t$ -links and  $n$ -links (red edges) for the central pixel and its neighbor pixels (right). In this example, there are 127 vertices and 350 edges.

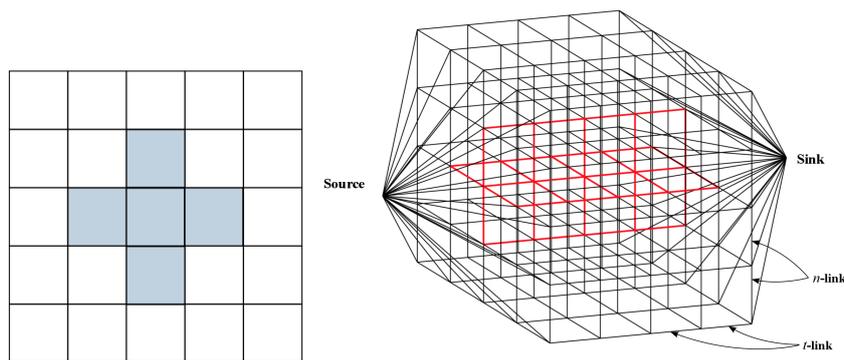


Figure 2. An image of size 5 × 5 pixels (left) and its graph topology (right).

A cut separating the source from the sink severs *t*-links as well as *n*-links. The severed *t*-links directly define the labels assigned to each pixel. For instance, if a cut severs the *t*-link  $t_{p_j} = \{p_j, p_{j+1}\}$  with  $0 \leq j \leq L$  for a pixel  $p$ , then the assigned label to  $p$  is  $j$ . Therefore, the cut must sever exactly one *t*-link at every pixel.

The cost of a *t*-link will correspond to the disparity of a given pixel. If the assumption that we have a certain disparity  $d$  at pixel  $p = (x_1, y)$  is correct, the intensity  $I_l(x, y)$  observed in the left camera at position  $(x, y)$  should be similar to the intensity  $I_r(x - d, y)$  in the right camera observed at position  $(x - d, y)$ , because both pixels depict the same scene object. Consequently, the cost of a given *t*-link  $t_{p_d} = \{p_d, p_{d+1}\}$  with  $0 \leq d \leq L$  can be set to the  $L_2$  norm of the intensity difference between corresponding pixels in both camera images:

$$c(t_{p_d}) = |I_l(x, y) - I_r(x - d, y)|^2 + C \tag{12}$$

where  $C$  is a constant that is chosen to be sufficiently large in order to ensure that exactly one *t*-link is severed by the cut for each pixel. If the intensity difference between  $I_l(x, y)$  and  $I_r(x - d, y)$  is low enough, this indicates that the corresponding disparity is a good solution and consequently,  $c(t_{p_d})$  is low. The constant  $C$  is set to be slightly larger than the sum of the weights of all penalty *n*-links belonging to pixel  $p$ ,

$$C_p = 1 + (L - 1) \sum_{q \in \mathcal{N}_p} \lambda_{pq} \tag{13}$$

where  $\mathcal{N}_p$  defines the neighborhood of  $p$ .

The pairwise cost function  $V_{pq}$  depends on the difference between labels  $l_p$  and  $l_q$ . In the simplest case,  $V_{pq}$  can be set to the absolute difference  $|l_p - l_q|$ . Therefore,

$$V_{pq}(l_p, l_q) = \lambda_{pq} |l_p - l_q| \tag{14}$$

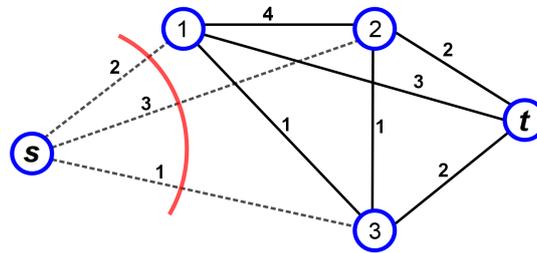
where  $\lambda_{pq}$  is a weighting factor for setting the relative importance of the smoothness term, which can be chosen differently for each pair of pixels, but is often set to a constant  $\lambda = \lambda_{pq}$  for all  $p, q \in \mathcal{N}$ . Given an *n*-link  $n_{p_k q_k}$  for two neighbor pixels  $p$  and  $q$ , the cost of  $n_{p_k q_k}$  is equal to  $c(n_{p_k q_k}) = \lambda_{pq}$  for all  $1 \leq k < L$ .

#### 2.1.4. QUBO Formulation of the Stereo Matching Problem via the Minimum Multicut Problem

The graph cut formulation of the stereo matching problem is a special case of the following general problem:

**Problem 2** (Minimum multi-cut). *Given an edge weighted graph  $G = (V, E, c)$  and a set  $S = \{(s_1, t_1), \dots, (s_k, t_k)\}$  of  $k$  pairs of vertices, find a multi-cut with minimum cost, i.e., a set  $E' \subseteq E$  such*

that the removal of  $E'$  from  $E$  disconnects  $s_i$  from  $t_i$  for every pair  $(s_i, t_i) \in S$ , where the cost of  $E'$  is given as  $\sum_{\{u,v\} \in E'} c(\{u,v\})$  (see Figure 3).



**Figure 3.** Example of a multi-cut with  $k = 1$  for the pair of vertices  $(s, t)$ . The dotted lines are the edges belonging to the minimum multi-cut and its cost is the sum of their weights.

When  $k = 1, 2$ , the minimum multi-cut problem can be solved in polynomial time [46], and it becomes NP-hard for  $k \geq 3$  for general graphs [47]. In the special case where  $G$  is restricted to trees and for arbitrary  $k$ , Problem 2 remains NP-hard [48].

Let us formulate Problem 2 for  $k = 1$  as a QUBO expression. Let  $G = (V, E, c)$  be a weighted graph and  $(s, t)$  be the pair of vertices to be disconnected. For each edge  $\{u, v\} \in E$ , let us associate a Boolean variable  $y_{uv}$  such that  $y_{uv} = 1$  if the edge  $\{u, v\}$  is selected for a cut and  $y_{uv} = 0$  otherwise. Similarly, for each vertex  $v \in V$ , let us associate a Boolean variable  $x_v$  such that  $x_v = 1$  if  $v \in U$  and  $x_v = 0$  in the other cases where  $U$  is a subset of  $V$ .

Let  $H_{\text{problem}}$  be defined as

$$H_{\text{problem}} = H_{\text{cost}} + H_{\text{penalty}}. \tag{15}$$

The first term,  $H_{\text{cost}}$ , in Equation (15) expresses the cost of the selected edges to be removed from the graph  $G$ , and the second term,  $H_{\text{penalty}}$ , has the purpose of introducing a penalization if the selected edges do not correspond to a cut in  $G$ .

The term  $H_{\text{cost}}$  can be easily defined as

$$H_{\text{cost}} = \sum_{\{u,v\} \in E} y_{uv} \cdot c(\{u,v\}) \tag{16}$$

which gives us the cost of the selected edges to be removed.

The second term,  $H_{\text{penalty}}$ , is constructed as

$$H_{\text{penalty}} = \alpha [1 - x_s - x_t + 2x_s x_t + \sum_{\{u,v\} \in E} (1 - y_{uv})(x_u + x_v - 2x_u x_v)]. \tag{17}$$

The penalty term satisfies the criterion that if the selected edges form a cut, then  $H_{\text{penalty}} = 0$ , and  $H_{\text{penalty}} > 0$  in other cases. The penalty term is constructed based on the observation that the selected edges form a cut if there exists a partition  $(U, \bar{U})$  with  $U \subseteq V$  such that  $s \in U$  and  $t \in \bar{U}$ , and every edge  $e \in E$  with  $y_e = 0$ , has its extreme vertices on only one side of the partition, either in  $U$  or in  $\bar{U}$ . The coefficient  $\alpha$  is a positive value which is used to ensure that solutions that do not constitute a cut are avoided.

The expression  $H_{\text{penalty}}$  given in (17) corresponds to a pseudo-Boolean function of the third degree which can be converted into a QUBO expression using a reduction method. Among reduction methods we find the Boros algorithms [37,49,50] and the Freedman [51] and Ishikawa [52] methods. A review of quadratization methods in pseudo-Boolean optimization can be found in [53].

The Ishikawa method says that every positive cubic term  $x_1x_2x_3$  can be expressed as a quadratic term with

$$\begin{aligned} x_1x_2x_3 &= x_1x_2 + x_1x_3 + x_2x_3 + \min_{z \in \{0,1\}} z(1 - x_1 - x_2 - x_3) \\ &\leq x_1x_2 + x_1x_3 + x_2x_3 + z(1 - x_1 - x_2 - x_3) \end{aligned} \quad (18)$$

where  $z$  is an ancilla Boolean variable. By applying the Ishikawa method to Equation (17) (which is the method that uses the smallest number of ancilla variables to obtain a quadratic expression) we have

$$\begin{aligned} H_{\text{penalty}}^{\text{qubo}} &= \alpha [1 - x_s - x_t + 2x_sx_t + \sum_{\{u,v\} \in E} (x_u + \\ &\quad x_v + 2w_{uv} + x_ux_v + x_vy_{uv} - 2x_ux_v - \\ &\quad 2x_vw_{uv} - 2y_{uv}w_{uv})]. \end{aligned} \quad (19)$$

where  $w_{uv}$  for each  $\{u, v\} \in E$  are ancilla variables.

The final quadratic cost function to be minimized can be written as

$$H_{\text{problem}}^{\text{qubo}} = H_{\text{cost}} + H_{\text{penalty}}^{\text{qubo}} \quad (20)$$

which requires a number  $2|E| + |V|$  of variables to represent the minimum multi-cut problem for a single pair of vertices.

### 3. Experiments and Discussion

In this section, we prove the concept of the solution to the stereo matching problem using the methodology described in Section 2. The following steps summarise our methodology:

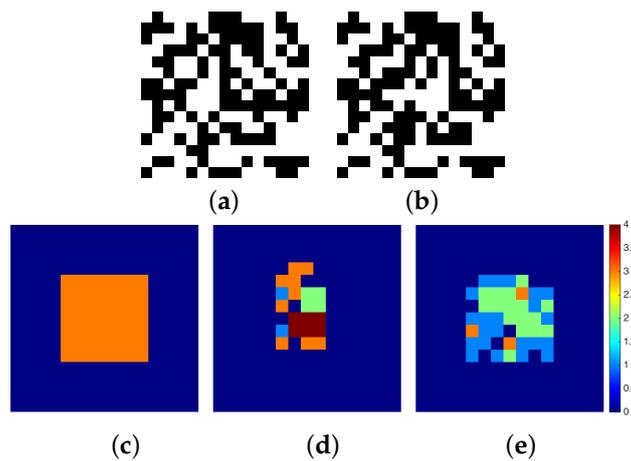
- (i) Construct the weighted graph described in Section 2.1.3 for the given stereo pair of images.
- (ii) Formulate the stereo matching problem as the minimum multi-cut problem for the pair  $(s, t)$  using the weighted graph from the previous step.
- (iii) Construct the QUBO expression given in Equation (20) from the weighted graph in step (ii) for a single pair.
- (iv) Embed into the Chimera graph topology of the D-Wave computer.
- (v) Find the minimum energy solution to the QUBO/Ising problem by quantum annealing.

We consider two case studies of stereo pair images: (a) a binary random dot stereogram (BRDS) and (b) a gray scale imaged object, both with a resolution of  $15 \times 15$  pixels. At this stage of our research, we are unable to study bigger size images because of the large resources needed to represent the stereo matching problem with the current limitations of quantum hardware. In case (a), we generate a BRDS as follows: a left image is created by setting each pixel randomly to either black or white. Then, a copy of this image is made and it is called the right image. Take a region in the right image and shift that region horizontally by  $d$  pixels to the left. The pixels vacated by shifting the region are filled in with random values. The shifted region simulates an object with a disparity of  $d$  pixels. In Figure 4 (top), we show a BRDS pair of images with a centered square region of size  $7 \times 7$  pixels shifted by  $d = 3$  pixels.

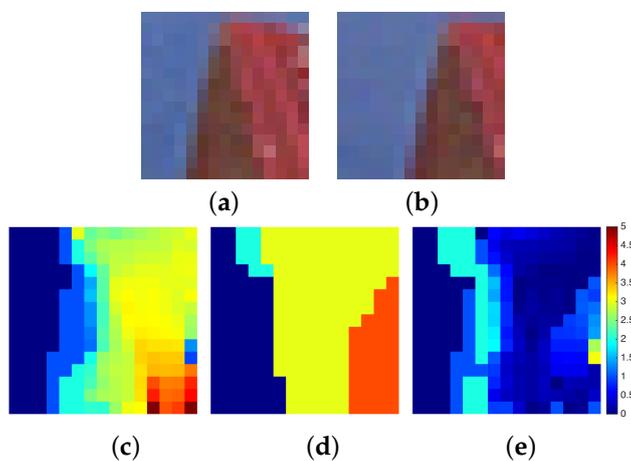
For case (b), we crop a region of size  $15 \times 15$  pixels from a stereo pair of images of a larger size. The cropped images have a maximum disparity range of five pixels. Figure 5 (top) shows a stereo pair showing a corner of an object on a background. In cases (a) and (b), the regions of interest (ROIs) to be matched are of sizes  $15 \times 12$  and  $15 \times 10$  pixels, respectively. In Table 1, the size of the weighted graph described in Section 2.1.3 for step (i) in our methodology is shown. Despite the large number of vertices and edges needed to formulate the stereo matching as a graph-cut problem, it is possible to use simplification techniques to limit the volume of the graph [45,54,55].

**Table 1.** For case studies (a) and (b) in Figures 4 and 5, respectively, the region of interest (ROI), the disparity range, the dimension of the weighted graph described in Section 2.1.3, and the number of logical variables used to formulate the minimum multi-cut problem given in Equation (19) are shown.

Case	ROI	L	V	E	$ H_{\text{problem}}^{\text{qubo}} $
(a)	$15 \times 12$	3	902	2745	6392
(b)	$15 \times 10$	5	1052	3125	7302

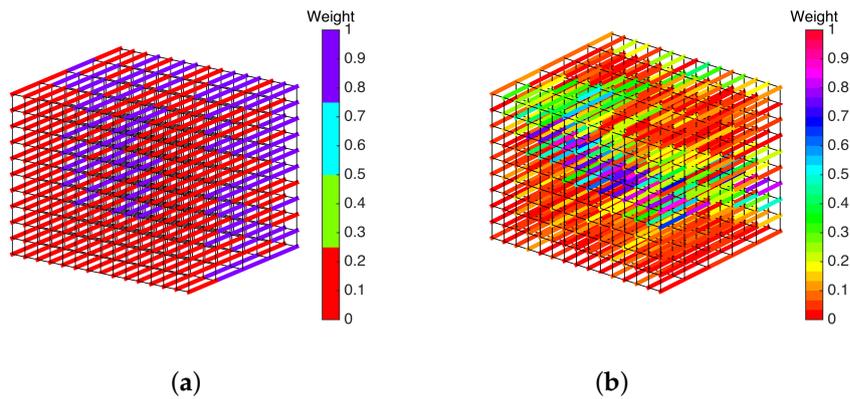


**Figure 4.** Example of binary random dot stereogram (BRDS) images of size  $15 \times 15$  pixels: (a,b) left and right BRDS images, (c) truth disparity map, (d) disparity map obtained using matching techniques, and (e) disparity map obtained using our approach.



**Figure 5.** Example of gray scale images of size  $15 \times 15$  pixels: (a,b) left and right gray scale stereo images, (c) disparity map using matching techniques with sub-pixel accuracy, (d) disparity map obtained using our approach, and (e) map error between the disparity maps (c,d).

Figure 6 shows the weighted graphs for cases (a) and (b) where the vertices  $s, t$  and their incident edges are omitted for better visualization. The  $t$ -links in the graph, represented by wide lines, are colored according to their weights, and the  $n$ -links are colored in black as thinner lines. As it can be seen in Figure 6a, we have only two possible weights for a  $t$ -link because the cost given in (12) for a BRDS can be zero or one plus a constant value.



**Figure 6.** Weighted graphs described in Section 2.1.3 for the (a) binary random dot stereogram BRDS and (b) gray-scale stereo pair images given in Figures 4 and 5, respectively.

In the center vertices of the weighted graph, we can see the square region to be matched. In Figure 6a, we show the weighted graph for the gray scale stereo pair given in Figure 5. In this case, there are many possible weights or disparity options. The weights of the omitted  $t$ -links in Figure 6 are set as high as possible so that they can never be chosen in the multi-cut. In cases (a) and (b), the constant weight  $\lambda$  is chosen in (14) which is equal to  $L$  for the  $n$ -links since the disparity is, at most,  $L - 1$ , and the constant value  $C$  in (12) is set to be equal to the maximum value of  $|I_l(x, y) - I_r(x - d, y)|^2$  for all possible values from  $d = 0, \dots, L - 1$  for all pixels.

In step (iii), the stereo matching problem is formulated as a graph-cut problem for a single pair  $(s, t)$ . For this purpose, a QUBO function is constructed as given in (20). Table 1 presents the number of logical Boolean variables needed to construct the QUBO function for cases (a) and (b). The QUBO function  $H_{\text{problem}}^{\text{qubo}}$  can be constructed for the given weighted graphs in Figure 6 as follows:

$$\begin{aligned}
 H_{\text{problem}}^{\text{qubo}} = & \sum_{\{u,v\} \in E_{tlink}} y_{uv}c(\{u,v\}) + \sum_{\{u,v\} \in E_{nlink}} y_{uv}c(\{u,v\}) + \alpha(1 - x_s - x_t + 2x_sx_t) + \\
 & \alpha \sum_{\{u,v\} \in E_{tlink}} (x_u + x_v + 2w_{uv} + x_u y_{uv} - 2x_u w_{uv} - 2x_v w_{uv} - 2y_{uv} w_{uv}) + \\
 & \alpha \sum_{\{u,v\} \in E_{nlink}} (x_u + x_v + 2w_{uv} + x_u y_{uv} - 2x_u w_{uv} - 2x_v w_{uv} - 2y_{uv} w_{uv}) \quad (21)
 \end{aligned}$$

where the set of edges  $E = E_{tlink} \cup E_{nlink}$  such that  $E_{tlink}$  is the set of  $t$ -links and  $E_{nlink}$  is the set of  $n$ -links. For the weighted graphs in Figure 6a,b, the number of  $n$ -links and  $t$ -links are  $|E_{nlink}| = 1080$ ,  $|E_{nlink}| = 1665$ , and  $|E_{nlink}| = 1200$ ,  $|E_{nlink}| = 1925$ , respectively.

The coefficient  $\alpha$  in (19) that is used to penalize sets of edges that do not correspond to a multi-cut is

$$\alpha = \sum_{\{u,v\} \in E} c(\{u,v\}) - \sum_{u \in V} [c(\{s,u\}) + c(\{t,u\})] \quad (22)$$

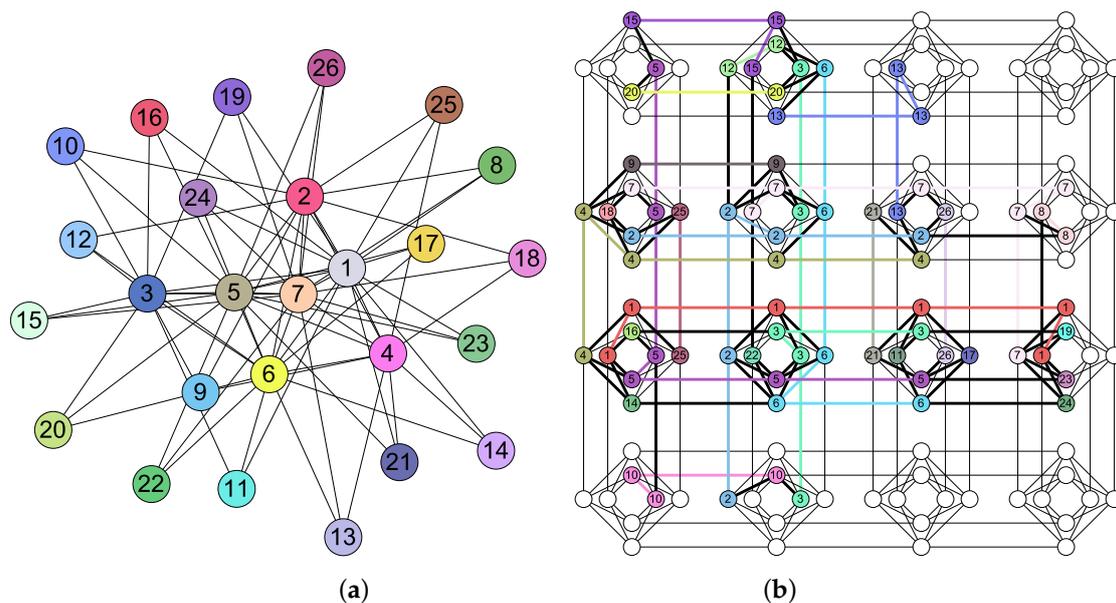
where the second term corresponds to the sum of the cost of all  $t$ -links that are adjacent to the source and sink vertices  $s$  and  $t$ . We omit the contribution of these  $t$ -links since they can never be selected for a multi-cut due to their high assigned cost.

The embedding procedure in step (iv) of the methodology consists of matching the connectivity of a given general graph into the D-Wave graph topology (see Figure 7). Translating a problem expressed as a QUBO function to an identical problem as a subgraph of the graph topology involves a process called minor embedding [56,57]. Commonly, the embedding process requires a number of physical qubits that is greater than the number of logical variables used in a given QUBO expression. Also, finding a minor embedding with the smallest number of physical qubits is an NP-hard problem [57].

The minor embedding problem can be stated as follows. Given the graph topology  $\mathcal{G}$  of the quantum hardware and a QUBO problem  $H_{\text{problem}}^{\text{qubo}}$ , represented as a logical graph  $G = (V, E)$ , where  $V$  corresponds to the set of Boolean variables in  $H_{\text{problem}}^{\text{qubo}}$  and  $E$  corresponds to the set of edges consisting of a pair of Boolean variables for each quadratic term in  $H_{\text{problem}}^{\text{qubo}}$ , find a subgraph in  $\mathcal{G}$  such that

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

Finding a minor embedding with the smallest number of physical qubits is an NP – hard problem [57]. Figure 7a shows a logical graph with 26 vertices corresponding a QUBO expression, and Figure 7b shows an embedding into the hardware topology of the given logical graph.



**Figure 7.** An example of minor embedding: (a) logical graph and (b) embedding of the logical graph into the hardware of the D-Wave One with 128 physical qubits.

We have simulated our algorithm using the software toolbox `qbsolv` that solves large QUBO problems by partitioning into subproblems targeted for execution on a D-Wave system [58]. Indeed, in addition to `qbsolv`, there are several QUBO solvers available, among them is [59,60] as well as those methods and software packages mentioned in [61]. We selected `qbsolv` because we are interested in testing our algorithm on D-Wave’s technology. Source code and full documentation of `qbsolv` can be found in [62,63].

Our experiments were run on `qbsolv` tool using a desktop computer MacBook Air with a 1.3 GHz Intel Core i5 processor and 4 GB of RAM. The motivation of using `qbsolv` is that if the D-Wave quantum hardware is available, then the problem is partitioned into smaller subproblems that can be minimized by executing a quantum search on the quantum hardware. On the other hand, if no hardware is available, then the `qbsolv` tool executes the classical Tabu search algorithm to solve each subproblem.

We compare the solutions obtained using `qbsolv` with a traditional block matching technique. In Figure 4c, we present the truth disparity map which consists of a square region of  $7 \times 7$  pixels with a depth of 3 pixels. For the BRDS stereo pair, the result using a block matching algorithm can be seen in Figure 4d, and the result obtained using `qbsolv` can be seen in Figure 4e. In the first case, the square region is partially reconstructed with high disparity values, and in the second case it is almost completely reconstructed with low disparity values. In the latter case, it takes 91.67 s of classic cpu time and 1288 calls using `qbsolv`. For the gray-scale stereo pair images in Figure 5, we compare

the result of a block matching algorithm with sub-pixel accuracy shown in Figure 5c with the result using qbsolv shown in Figure 5d. Figure 5e shows the absolute difference error between the disparity maps in Figure 5c,d, where the mean absolute error is 0.7359 pixels. In this case, it takes 148.31 s of classic cpu time and 1426 calls using qbsolv.

The examples presented in this section were selected with the purpose of showing how our proposal could be implemented. Due to hardware constraints, images used in this section are composed of a limited number of bits. We leave for future work a scalability analysis of our algorithm as well as a comparison study of our algorithm with its classical counterparts.

#### 4. Conclusions

We have presented a method for rewriting the stereo matching problem as QUBO expressions based on the graph-cut and minimum multi-cut problems and the Ishikawa reduction method, followed by using QUBO expressions to simulate quantum annealing algorithms on D-Wave's simulation software. This method has been validated using BRDS and gray scale stereo pairs of images. Future research will be focused on reducing the number of vertices in the graph-cut formulation of the stereo matching problem. This can be achieved by doing an initial disparity search to bound the limits of the disparity range for every pixel of the reference image as well as by incorporating an occlusion term in the cost model given in (11) to penalize occluded regions. An important contribution of this research is the quantum formulation of the minimum multi-cut problem for the case of a single pair of vertices. This formulation consists of the construction of a QUBO expression that is equivalent to the graph-cut problem; hence, no reduction algorithms are needed.

Although the proposed problem can be efficiently solved in a classical way, our methodology provides us with a new way of dealing with problems using quantum-annealing based algorithms and quantum technologies, such as the D-Wave annealer processor, to solve tasks within the realm of machine learning and related areas.

**Author Contributions:** W.C.-S. proposed the idea of writing a quantum annealing algorithm for the stereo matching problem. W.C.-S., S.E.V.-A., and M.L. made contributions towards writing and simulating the stereo matching problem in terms of a quantum annealing algorithm. W.C.-S., S.E.V.-A., and M.L. reviewed the manuscript.

**Funding:** SEVA 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).

**Acknowledgments:** S.E.V.-A. gratefully thanks his family for their unconditional support.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Klette, R. *Concise Computer Vision: An Introduction into Theory and Algorithms*; Springer Publishing Company: London, UK, 2014.
2. Stockman, G.; Shapiro, L.G. *Computer Vision*, 1st ed.; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2001.
3. Szeliski, R. *Computer Vision: Algorithms and Applications*, 1st ed.; Springer: New York, NY, USA, 2010.
4. Brown, M.Z.; Burschka, D.; Hager, G.D. Advances in Computational Stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 993–1008. [[CrossRef](#)]
5. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: New York, NY, USA, 2003.
6. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle Adjustment—A Modern Synthesis. In Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, Corfu, Greece, 21–22 September 2000; Springer: London, UK, 2000; pp. 298–372.
7. Darrell, T.; Gordon, G.; Harville, M.; Woodfill, J. Integrated Person Tracking Using Stereo, Color, and Pattern Detection. *Int. J. Comput. Vis.* **2000**, *37*, 175–185. [[CrossRef](#)]

8. Seitz, S.M.; Curless, B.; Diebel, J.; Scharstein, D.; Szeliski, R. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; IEEE Computer Society: Washington, DC, USA, 2006; Volume 1, pp. 519–528. [[CrossRef](#)]
9. Uyttendaele, M.; Criminisi, A.; Kang, S.B.; Winder, S.; Szeliski, R.; Hartley, R. Image-Based Interactive Exploration of Real-World Environments. *IEEE Comput. Graph. Appl.* **2004**, *24*, 52–63. [[CrossRef](#)] [[PubMed](#)]
10. Agrawal, M.; Konolige, K.; Bolles, R.C. Localization and Mapping for Autonomous Navigation in Outdoor Terrains: A Stereo Vision Approach. In Proceedings of the 2007 IEEE Workshop on Applications of Computer Vision (WACV '07), Austin, TX, USA, 21–22 February 2007; p. 7. [[CrossRef](#)]
11. Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2014**, *56*, 172–185. [[CrossRef](#)]
12. Biamonte, J.D.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [[CrossRef](#)] [[PubMed](#)]
13. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631–633. [[CrossRef](#)]
14. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [[CrossRef](#)] [[PubMed](#)]
15. Adachi, S.H.; Henderson, M.P. Application of Quantum Annealing to Training of Deep Neural Networks. *arXiv* **2015**, arXiv:1510.06356.
16. Kulchytskyy, B.; Andriyash, E.; Amin, M.; Melko, R. Quantum Boltzmann Machine. *arXiv* **2016**, arXiv:1601.02036.
17. Benedetti, M.; Realpe-Gómez, J.; Biswa, R.; Perdomo-Ortiz, A. Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Phys. Rev. A* **2016**, *94*, 022308. [[CrossRef](#)]
18. Benedetti, M.; Realpe-Gómez, J.; Biswa, R.; Perdomo-Ortiz, A. Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models. *Phys. Rev. X* **2017**, *7*, 130503. [[CrossRef](#)]
19. IBM Quantum Experience. Available online: <http://research.ibm.com/ibm-q/> (accessed on 7 January 2018).
20. D-Wave systems. Available online: <http://www.dwavesys.com/> (accessed on 7 January 2018).
21. Perdomo-Ortiz, A.; Dickson, N.; Drew-Brook, M.; Rose, G.; Aspuru-Guzik, A. Finding low-energy conformations of lattice protein models by quantum annealing. *Sci. Rep.* **2012**, *2*, 571. [[CrossRef](#)] [[PubMed](#)]
22. Perdomo-Ortiz, A.; Fluegemann, J.; Narasimhan, S.; Biswas, R.; Smelyanskiy, V. A quantum annealing approach for fault detection and diagnosis of graph-based systems. *Eur. Phys. J. Spec. Top.* **2015**, *224*, 131–148. [[CrossRef](#)]
23. King, J.; Yarkoni, S.; Nevisi, M.M.; Hilton, J.P.; McGeoch, C.C. Benchmarking a quantum annealing processor with the time-to-target metric. *arXiv* **2015**, arXiv:quant-ph/1508.05087.
24. Bian, Z.; Chudak, F.; Israel, R.B.; Lackey, B.; Mcready, W.G.; Roy, A. Mapping Constrained Optimization Problems to Quantum Annealing with Application to Fault Diagnosis. *Front. ICT* **2016**, *3*, 14. [[CrossRef](#)]
25. Tsukamoto, S.; Takatsu, M.; Matsubara, S.; Tamura, H. An Accelerator Architecture for Combinatorial Optimization Problems. *Fujitsu Sci. Tech. J.* **2017**, *53*, 8–13.
26. Yamaoka, M.; Yoshimura, C.; Hayashi, M.; Okuyama, T.; Aoki, H.; Mizuno, H. Ising Computer. *Hitachi Rev.* **2016**, *65*, 78–82.
27. Venegas-Andraca, S.; Cruz-Santos, W.; McGeoch, C.; Lanzagorta, M. A cross-disciplinary introduction to quantum annealing-based algorithms. *Contemp. Phys.* **2018**, *5982*, 174–196. [[CrossRef](#)]
28. Apolloni, B.; Cesa-Bianchi, N.; de Falco, D. A numerical implementation of Quantum Annealing. In Proceedings of the Stochastic Processes, Physics and Geometry, Ascona, Switzerland, 4–9 July 1988; pp. 97–111.
29. Apolloni, B.; Carvalho, C.; de Falco, D. Quantum stochastic optimization. *Stoch. Process. Appl.* **1989**, *33*, 233–244. [[CrossRef](#)]
30. Somorjai, R. Novel approach for computing the global minimum of proteins. 1. General concepts, methods, and approximations. *J. Phys. Chem.* **1991**, *95*, 4141–4146. [[CrossRef](#)]
31. Amara, P.; Hsu, D.; Straub, J.E. Global energy minimum searches using an approximate solution of the imaginary time Schroedinger equation. *J. Phys. Chem.* **1993**, *97*, 6715–6721. [[CrossRef](#)]

32. Finnila, A.; Gomez, M.; Sebenik, C.; Stenson, C.; Doll, J. Quantum annealing: A new method for minimizing multidimensional functions. *Chem. Phys. Lett.* **1994**, *219*, 343–348. [[CrossRef](#)]
33. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **1998**, *58*, 5355–5363. [[CrossRef](#)]
34. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]
35. Santoro, G.E.; Tosatti, E. Optimization using quantum mechanics: quantum annealing through adiabatic evolution. *J. Phys. A Math. Gen.* **2006**, *39*, R393–R431. [[CrossRef](#)]
36. Barahona, F. On the computational complexity of Ising spin glass models. *J. Phys. A Math. Gen.* **1982**, *15*, 3241. [[CrossRef](#)]
37. Boros, E.; Hammer, P.L. Pseudo-boolean Optimization. *Discret. Appl. Math.* **2002**, *123*, 155–225. [[CrossRef](#)]
38. Farhi, E.; Goldstone, J.; Gutmann, S.; Lapan, J.; Lundgren, A.; Preda, D. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science* **2001**, *292*, 472–475. [[CrossRef](#)] [[PubMed](#)]
39. Jansen, S.; Ruskai, M.B.; Seiler, R. Bounds for the adiabatic approximation with applications to quantum computation. *J. Math. Phys.* **2007**, *48*, 102111. [[CrossRef](#)]
40. Forsyth, D.A.; Ponce, J. *Computer Vision: A Modern Approach*; Pearson: New York, NY, USA, 2002.
41. Ohta, Y.; Kanade, T. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE Trans. Pattern Anal. Mach. Intell.* **1985**, *PAMI-7*, 139–154. [[CrossRef](#)]
42. Sun, J.; Zheng, N.N.; Shum, H.Y. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 787–800. [[CrossRef](#)]
43. Boykov, Y.; Veksler, O.; Zabih, R. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [[CrossRef](#)]
44. Boykov, Y.; Kolmogorov, V. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [[CrossRef](#)] [[PubMed](#)]
45. Veksler, O. Reducing Search Space for Stereo Correspondence with Graph Cuts. In Proceedings of the British Machine Vision Conference 2006, Edinburgh, UK, 4–7 September 2006; pp. 709–718.
46. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1982.
47. Dahlhaus, E.; Johnson, D.S.; Papadimitriou, C.H.; Seymour, P.D.; Yannakakis, M. The Complexity of Multiterminal Cuts. *SIAM J. Comput.* **1994**, *23*, 864–894. [[CrossRef](#)]
48. Garg, N.; Vazirani, V.; Yannakakis, M. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica* **1997**, *18*, 3–20. [[CrossRef](#)]
49. Boros, E.; Gruber, A. On Quadraticization of Pseudo-Boolean Functions. *arXiv* **2014**, arXiv:1404.6538.
50. Anthony, M.; Boros, E.; Crama, Y.; Gruber, A. Quadraticization of symmetric pseudo-Boolean functions. *Discret. Appl. Math.* **2016**, *203*, 1–12. [[CrossRef](#)]
51. Freedman, D.; Drineas, P. Energy Minimization via Graph Cuts: Settling What is Possible. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, 20–26 June 2005; pp. 939–946. [[CrossRef](#)]
52. Ishikawa, H. Transformation of General Binary MRF Minimization to the First-Order Case. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1234–1249. [[CrossRef](#)] [[PubMed](#)]
53. Tanburn, R.; Dattani, N.S. Quadraticization in Pseudo-Boolean Optimization and Adiabatic Quantum Computing. Available online: <https://github.com/ndattani/quadraticizationReview> (accessed on 17 September 2018).
54. Zureiki, A.; Devy, M.; Chatila, R. Stereo Matching using Reduced-Graph Cuts. In Proceedings of the 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 16 September–19 October 2007; Volume 1, pp. 237–240. [[CrossRef](#)]
55. Zhang, Y.; Hartley, R.I.; Wang, L. Fast Multi-labelling for Stereo Matching. In Proceedings of the 11th European Conference on Computer Vision (ECCV 2010), Heraklion, Crete, Greece, 5–11 September 2010; pp. 524–537.
56. Choi, V. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Inf. Process.* **2008**, *7*, 193–209. [[CrossRef](#)]

57. Choi, V. Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quantum Inf. Process.* **2011**, *10*, 343–353. [[CrossRef](#)]
58. Booth, M.; Reinhardt, S.P.; Roy, A. *Partitioning Optimization Problems for Hybrid Classical/Quantum Execution*; Technical Report; D-Wave the Quantum Computing Company: Burnaby, CA, USA, 9 January 2017.
59. IBM ILOG CPLEX Optimizer. Available online: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer> (accessed on 14 September 2018).
60. METSlib An Open Source Tabu Search Metaheuristic framework in modern C++. Available online: <http://projects.coin-or.org/metslib> (accessed on 14 September 2018).
61. Rosenberg, G.; Vazifeh, M.; Woods, B.; Haber, E. Building an iterative heuristic solver for a quantum annealer. *Comput. Optim. Appl.* **2016**, *65*, 845–869. [[CrossRef](#)]
62. Qbsolv Software, Version 2000Q. D-Wave Systems. Burnaby, CA, USA, 2000. Available online: <https://www.dwavesys.com/software> (accessed on 14 September 2018).
63. D-Wave sOource Code and Documentation. Available online: <https://github.com/dwavesystems/qbsolv> (accessed on 14 September 2018).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).