



Received June 29, 2021; revised January 27, 2022; accepted February 9, 2022; date of publication February 25, 2022;  
date of current version May 10, 2022.

Digital Object Identifier 10.1109/TQE.2022.3153947

# Quantum Annealing Methods and Experimental Evaluation to the Phase-Unwrapping Problem in Synthetic Aperture Radar Imaging

KHALED A. HELAL KELANY<sup>1</sup> ,  
NIKITAS DIMOPOULOS<sup>1</sup>  (Life Senior Member, IEEE),  
CLEMENS P. J. ADOLPHS<sup>2</sup>, AND AMIRALI BANIASADI<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada

<sup>2</sup>IQB Information Technologies, Vancouver, BC V6E 4B1, Canada

Corresponding author: Khaled A. Helal Kelany (e-mail: khaledkelany@uvic.ca).

This work was supported in part by a MITACS accelerate internship and in part by NSERC through the Discovery grants program.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TQE.2022.3153947>, provided by the authors.

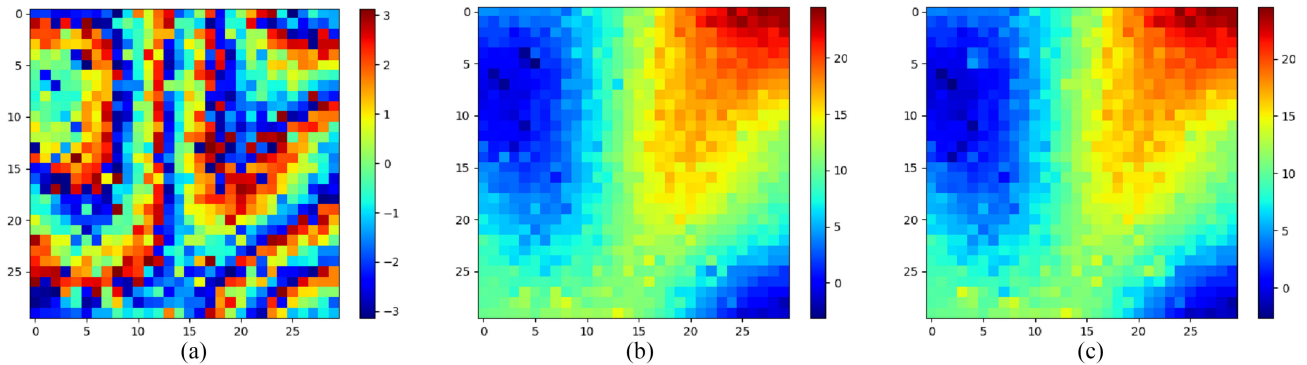
**ABSTRACT** The focus of this work is to explore the use of quantum annealing solvers for the problem of phase unwrapping of synthetic aperture radar (SAR) images. Although solutions to this problem exist based on network programming, these techniques do not scale well to larger sized images. Our approach involves formulating the problem as a quadratic unconstrained binary optimization (QUBO) problem, which can be solved on a quantum annealer. Given that present embodiments of quantum annealers remain limited in the number of qubits they possess, we decompose the problem into a set of subproblems that can be solved individually. These individual solutions are close to optimal up to an integer constant, with one constant per subimage. In a second phase, these integer constants are determined as a solution to yet another QUBO problem. This basic idea is extended to several passes, where each pass results in an image which is subsequently decomposed to yet another set of subproblems until the resulting image can be accommodated by the annealer at hand. Additionally, we explore improvements to the method by decomposing the original image into overlapping subimages and ignoring the results on the overlapped (marginal) pixels. We test our approach with a variety of software-based QUBO solvers and on a variety of images, both synthetic and real. Additionally, we experiment using D-wave systems' quantum annealer, the D-wave 2000Q\_6 and developed an embedding method which, for our problem, yielded improved results. Our method resulted in high quality solutions, comparable to state-of-the-art phase-unwrapping solvers.

**INDEX TERMS** Interferometric synthetic aperture radar (SAR), phase unwrapping, quadratic unconstrained binary optimization (QUBO), quantum annealing.

## I. INTRODUCTION

Two-dimensional phase unwrapping is the process of recovering unambiguous phase values from a two-dimensional array of phase values known only modulo  $2\pi$  rad. The measured phase is also affected by random noise and systematic distortions. This problem arises when the phase is used as a proxy indicator of a physical quantity, which is the time delay between two signals in the case of interferometric synthetic aperture radar (InSAR) [4]. This time delay is significant, as it is affected by the height differences of

the illuminated target. It can thus be used to extract accurate 3-D topography and reveal topographical changes that occur over time. As the phase is observable only on a circular space, where all measured values are mapped to the range  $(-\pi, \pi]$ , the observed data must be mapped back to the full range of real phase values to be meaningful. To illustrate the concept, Fig. 1 shows a small wrapped synthetic image, the corresponding original one (the ground truth) and the unwrapped image obtained using an industry-standard unwrapping method based on the TRWS algorithm [3], [12].



**FIGURE 1.** Sample of a synthetic image unwrapped using the TRWS algorithm. The color bars on the right side of the figures represent the phase value in radian. On the left the wrapped image, ranging in value from  $-\pi$  to  $\pi$ . In the middle the ground truth image to compare against, ranging in value from  $-\pi$  to  $8\pi$ . On the right the unwrapped image, ranging in value from  $-\pi$  to  $8\pi$ .

For unwrapping purposes, the sampling rate is typically assumed to be suitable for most datasets to prevent aliasing, that is, the absolute difference in phase between two adjacent data points is assumed to be smaller than  $\pi$  (the Nyquist criterion [2]). This phase unwrapping problem represents a class of imaging techniques, such as InSAR, magnetic resonance imaging, and optical interferometry.

The development of InSAR and many other applications have stimulated interest in building accurate two-dimensional phase unwrapping algorithms. The most commonly used unwrapping technique is based on network programming strategies that formulate the problem as a minimum cost flow (MCF) [3] problem. One of these solvers is the sequential tree-reweighted message passing (TRWS) algorithm [12]. However, since the InSAR images can be quite large—normally larger than 600 M pixels—the process of phase unwrapping via TRWS can take a prohibitively long time on a classical computer. Hence, we explore whether a quantum computing system could be a potential candidate for solving the phase unwrapping problem.

Quantum computing exploits the laws of quantum mechanics to process information [13]. In contrast to classical computers, which use bits to process information, quantum computers use quantum bits, or qubits, as the basic units of quantum information. Analogously to bits, qubits encode state information. Qubits may be in either of the two distinct states of  $|0\rangle$  or  $|1\rangle$ , but they may also encode a superposition of these states, (i.e.,  $\alpha|0\rangle + \beta|1\rangle$ ) with complex-valued coefficients  $\alpha$  and  $\beta$ .

Quantum annealing is a quantum computing method used to find the optimal solution to certain combinatorial optimization problems [5]. This is achieved by using properties of quantum mechanics, such as quantum tunneling, entanglement, and superposition.

Quantum annealing systems are able to solve problems in quadratic unconstrained binary optimization (QUBO) form. Any unconstrained quadratic integer problem with bounded integer variables can be transformed by a binary expansion into QUBO form [7]. The phase unwrapping problem is a

quadratic unconstrained problem by default, and it can be mapped to a QUBO problem by simply encoding each variable [e.g.,  $k_t$  in (6)] into a vector of binary variables.

Because of limitations in accessing actual quantum annealing infrastructure, we have tested our methodology using a variety of QUBO solvers. As we will show, the results we obtain match the results obtained using the classical network optimization method (i.e., the TRWS method), which is considered the benchmark in addressing the unwrapping problem.

InSAR images tend to be quite large, often exceeding  $20 \times 30$  K, or 600-M, pixels. In the simplest problem, where each pixel label would require one qubit, a 600-M-qubit quantum annealer would be required; such a machine is not currently available. To overcome the limitations of present day technology, we have developed a method, where we partition the image and then use quantum annealing on the individual partitions to obtain suboptimal labeling, and then use quantum annealing in a second phase to obtain labels that approach the ones obtained through classical methods. We have named our method “super-pixel decomposition.”

The concept of the superpixel decomposition can be extended to handle bigger images by recursively partitioning the image until the partitions fit in the quantum annealer. Then recursively apply the second phase of the superposition until we obtain labels that approach the global solution. We name this method “multipass superpixel decomposition.”

The results can be further enhanced by utilizing additional (marginal) pixels in each of the subimages processed in the first phase of the method.

This work is an extension and refinement of the work we presented in [11]. In this work, we are presenting the results of the superpixel decomposition method, the single [11], and the multipass approach. In addition, we study the effect of the subimage size on the quality of the unwrapping. Finally, we are examining the addition of a margin to the subimage.

The rest of the article is organized as follows: In Section II we provide a background, in Section III we explain our methodology. In Sections IV–VI, we present the experimental results. Sections IV-A–C present the image databases we

used, the metrics we employed to determine the efficacy of the various methods and the solvers we employed including D-waves 2000Q\_6 quantum annealer. Section V presents the experimental results of the efficacy of the superpixel method as is affected by the noise levels and the solver chosen, while Section VI presents the experimental results of enhancing the superpixel methods through including marginal pixels when processing the subimages of the superpixel method. Section VII concludes this article.

## II. BACKGROUND

### A. PHASE UNWRAPPING FORMULATION

Strictly speaking, phase unwrapping is an ill-posed problem as the unwrapped phase array contains information that is not available in the wrapped array. Therefore, to perform correctly, all phase unwrapping methods rely on regularizing assumptions. The most common of these assumptions is that the Nyquist criterion is met throughout most (but not necessarily all) of the scene; that is, the spatial sampling rate is assumed to be high enough that aliasing is avoided [2].

The Nyquist criterion implies that the difference between the phases of two neighboring pixels is less than  $\pi$ . The key to phase unwrapping, therefore, lies not on directly calculating the unwrapped phase values themselves, but in estimating these values given that the differences of the wrapped phases are the same as those of the unwrapped phases dictated by the Nyquist assumption.

Let  $\phi$ ,  $\varphi$ , and  $k$  denote the unwrapped phase, the wrapped phase, and an integer label to be estimated, respectively. For the phase of a pixel  $i$ , we have

$$\phi_i = \varphi_i + 2\pi k_i. \quad (1)$$

The unwrapping problem can then be expressed as an optimization problem of the cost function

$$E(k) = \sum_{(s,t) \in A} W_{st} |k_t - k_s - a_{st}| \quad (2)$$

that is

$$\arg \max_k E(k) \quad (3)$$

where  $k_i$  are the labels that will determine the original phase as per (1);  $A$  is the set of pixels in the SAR image;  $W_{st}$  are weights defining the neighborhood structure; and  $a_{ij}$  are constants obtained from the image as per the equation

$$a_{ij} \stackrel{\text{def}}{=} \frac{\text{wrap}(\phi_i - \phi_j) - (\phi_i - \phi_j)}{2\pi} \quad (4)$$

where

$$\text{wrap}(\theta) = \arg(e^{i\theta}) = \theta - \left\lfloor \frac{\theta}{2\pi} \right\rfloor. \quad (5)$$

Appendix A details the derivation of the cost function shown in (2).

The optimization problem, as defined in (2), above admits several solutions since only the difference of the labels is

used in the cost function. Labels can be increased or decreased by the same amount and still result in the same minimal cost. A way to further regularize the solution is to insist that the desirable solution involves labels that are the smallest possible. Therefore, the cost function is augmented with an extra term that depends on the labels themselves as follows:

$$E = \sum_{(s,t) \in A} W_{st} |k_t - k_s - a_{st}| + \sum_{s \in A} \omega_s |k_s - a_s|. \quad (6)$$

The weights  $W_{st}$ ,  $\omega_s$ , and the bias  $a_s$  are chosen heuristically and represent ad-hoc information one may have on the scene represented in the image; most often  $a_s = 0$ .

Without loss of generality, one can also consider cost functions involving quadratic expressions of the labels instead of the more challenging absolute value ones

$$E = \sum_{(s,t) \in A} W_{st} (k_t - k_s - a_{st})^2 + \sum_{s \in A} \omega_s (k_s - a_s)^2 \quad (7)$$

and in case that  $a_s = 0$ , then a similar cost function is

$$E = \sum_{(s,t) \in A} W_{st} (k_t - k_s - a_{st})^2 + \sum_{s \in A} \omega_s k_s^2. \quad (8)$$

Although the  $L1$ -norm is preferred over the  $L2$ -norm in the continuous case—as the  $L2$ -norm tends to spread the error and does not result in good solutions [3]—this is not a factor in the integer case. The most commonly used method of solving the phase unwrapping problem (TRWS) is attributed to Kolmogorov [12].

### B. QUANTUM ANNEALING

Quantum annealing employs the quantum tunneling effect to ensure that a system is able to escape local minima as it traverses the state space of an energy function toward its way to ground-state settlement. The quantum computational systems, such as the ones by D-wave, use quantum annealing to locate the ground-state of an artificial Ising system [9]. An Ising Hamiltonian describes the behavior of, such a system as

$$H_p = \sum_{i=1}^N h_i \sigma_i^z + \sum_{i,j=1}^N J_{ij} \sigma_i^z \sigma_j^z \quad (9)$$

where  $h_i$  is the energy bias for spin  $i$ ;  $J_{ij}$  is the coupling energy between spins  $i$  and  $j$ ;  $\sigma_i^z$  is the Pauli spin matrix; and  $N$  is the number of qubits. Quantum annealing on this system is achieved by the gradual evolution of the Hamiltonian system [9]

$$H(t) = \Gamma(t) \sum_{i=1}^N \Delta_i \sigma_i^x + \Lambda(t) H_p. \quad (10)$$

As time passes,  $\Gamma$  decreases from 1 to 0, while  $\Lambda$  increases from 0 to 1. If the annealing is performed slowly enough, the system stays in the ground state of  $H(t)$  for all times,  $t$ , ending up at the end of the annealing at the ground state of  $H_p$ . The Hamiltonian in (9) can be rewritten

in vector form as  $H(s) = s^T J s + s^T h$ , in the form of a QUBO problem [16].

As used in the rest of this article, the objective function is expressed in QUBO form in scalar notation, and is defined as follows:

$$C(x) = \sum_i a_i x_i + \sum_{i < j} b_{i,j} x_i x_j \quad (11)$$

where  $x \in \{0; 1\}^n$  is a vector of binary variables and  $\{a_i; b_{i,j}\}$  are real coefficients.

Before an application problem can be solved on a quantum annealer, it must first be mapped into QUBO form. As a first step in transforming the InSAR problem into a QUBO problem, the  $k_i$  label that is nonbinary valued must be transformed into binary valued. Let  $k_i \in \{0, D_i - 1\}$ , where  $D_i$  is the number of allowed values (labels) for  $k_i$ . This can be achieved by writing  $k_i$  in binary. The binary transformation restricts the number of new-valued binary variables required to represent  $k_i$ . Let  $d_i = \lceil \log_2 D_i \rceil$  and  $k_i = \langle \mathbf{2}, \mathbf{x}_i \rangle$ , where the vector  $\mathbf{x}_i = [x_{i,d_i}, \dots, x_{i,1}, x_{i,0}]$  represents the bits of  $k_i$  and  $\mathbf{2} = [2^{d_i}, \dots, 2, 1]$  is the vector of powers of two. Equation (7) can be written in QUBO form as

$$E = \sum_{(s,t) \in A} W_{st} \left( \sum_i b_i x_{i,t} - \sum_i b_i x_{i,s} - a_{st} \right)^2 + \sum_{s \in A} \omega_s \left( \sum_i b_i x_{i,s} - a_s \right)^2 \quad (12)$$

where  $b_i$  is the weighting coefficient for the binary variable  $x_i$  ( $b_i = 2^i$  in the case of the binary encoding).

Many problems can be formulated to take advantage of quantum annealing, and is advantageous because it converges faster than other techniques to an optimum solution [10].

Quantum annealing can be compared to simulated annealing by identifying that the temperature parameter in simulated annealing performs a similar role to quantum tunneling in quantum annealing. The temperature in simulated annealing defines the probability of moving from a single current state to a higher energy state to escape local minima. The assumed advantage of quantum annealing over simulated annealing is that tunneling allows the system to directly pass through high energy barriers without having to climb over them.

Analytical and numerical evidence indicates that quantum annealing can outperform simulated annealing [8]. Therefore, quantum annealing is a good potential solver for the InSAR phase unwrapping problem.

## C. OPTIMIZERS

As we mentioned earlier, the size of the problem does not allow the direct use of currently available annealing infrastructure. Similarly, the size of the problem results in a prohibitively expensive QUBO computation if we elect to perform a global optimization on the full scale image. Rather, our methodology partitions the image, and QUBO solvers are applied first on the partitions and then in a second phase on an abstraction of the image comprised of what we call superpixels, each one representing a partition of the original image.

In Sections II-C1 and II-C2, we shall discuss the QUBO solvers we have employed, and then our approach of partitioning the image and the superpixel methodology.

### 1) CLASSICAL OPTIMIZER

#### a) TRWS [12]

The TRWS algorithm is used for discrete energy minimization, where the energy function can be formulated as follows:

$$E(x|\theta) = \theta_{\text{const}} + \sum_{s \in \nu} \theta_s(x_s) + \sum_{(s,t) \in \varepsilon} \theta_{st}(x_s, x_t) \quad (13)$$

where  $\nu$  corresponds to the set of pixels;  $x_s$  indicates the label of pixel  $s \in \nu$ ;  $\varepsilon$  corresponds to the set of edges (each edge connects two related pixels);  $\theta_s(\cdot)$  is the penalty function (i.e., a term of an unconstrained objective function added to add some constraint to it) of unary data; and  $\theta_{st}(\cdot, \cdot)$  is the penalty function of the pairwise terms. This energy function is usually derived in the context of Markov random fields [6]. TRWS is a message-passing [15] algorithm for finding optimal maximum *a posteriori* (MAP) configurations for a Markov random field represented as a factor graph. The algorithm is widely used in phase unwrapping problems, where the unary penalty functions represent the unary terms in (7), where the pixels are penalized for having large values, while the pairwise penalty functions represent the pairwise terms in (7), where the two pixels  $k_t$  and  $k_s$  are penalized for having a difference not equal to  $a_{st}$ .

### 2) QUBO OPTIMIZER

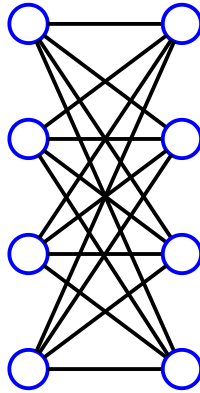
#### a) Microsoft Quantum-Inspired Optimization (MQIO)

QIO is a classical computing method. It refers to a class of algorithms inspired by quantum computing that are applied to solve optimization problems on traditional hardware. Microsoft QIO supports four solvers: 1) simulated annealing, 2) parallel tempering, 3) tabu search, and 4) quantum Monte Carlo. We are using the two solvers simulated annealing and parallel tempering.

#### b) Simulated Annealing

This solver provides an implementation of the simulated annealing method [14]. The solver is also one of the Microsoft QIO solvers. Therefore, we used this solver on small images for the same reasons mentioned above.





**FIGURE 2.** Chimera architecture unit cell. The circles represent qubits, and the edges represent couplers.

### c) Parallel Tempering

This solver is one of Microsoft’s quantum inspired optimization (Microsoft QIO) solvers and accessible through a cloud-client interface. However, we have limited early access to this solver. Hence, we used it only on small images.

### d) D-Wave Annealing

D-wave systems provides implementations of different quantum annealing systems, starting from the D-wave one announced in 2011 [9]. We used the D-wave 2000Q\_6 machine to unwrap the InSAR subimages. The machine contains 2041 qubits. The qubits are sparsely connected in an architecture known as a “Chimera” graph. The Chimera architecture comprises sets of connected unit cells. Each unit cell has two columns of four vertical qubits that are connected via couplers. All qubits in one column are connected to all qubits in the other column. However, the qubits within a column are not connected (see Fig. 2). Unit cells are tiled horizontally and vertically with adjacent qubits connected. The qubits are logically mapped into a matrix of  $16 \times 16$  unit cells, with eight qubits per cell. Each qubit is connected to six neighbors in the Chimera topology, four connections to the other qubits within the cell and two connections to the horizontally adjacent cells (or vertically adjacent cells if the qubit is in the left column). In theory, the Chimera architecture comprises  $16 \times 16 \times 8 = 2048$  qubits. In practice, however, the largest number of embeddable qubits is slightly smaller (2041 qubits) due to missing, or faulty, qubits, an issue that arises during manufacturing. This also results in there being some nonexistent connections.

Solving a QUBO problem on a quantum annealer requires embedding (mapping) each binary variable to one physical qubit or multiple chained (connected) qubits and mapping the quadratic coefficients of the objective function (in a QUBO form) onto couplers. Couplers provide the connections between the qubits that form the Chimera graph as explained previously. Couplers are “programmable,” i.e., their strength

can be adjusted to reflect the coefficients of the cost function. D-wave uses superconducting loops to implement the couplers.

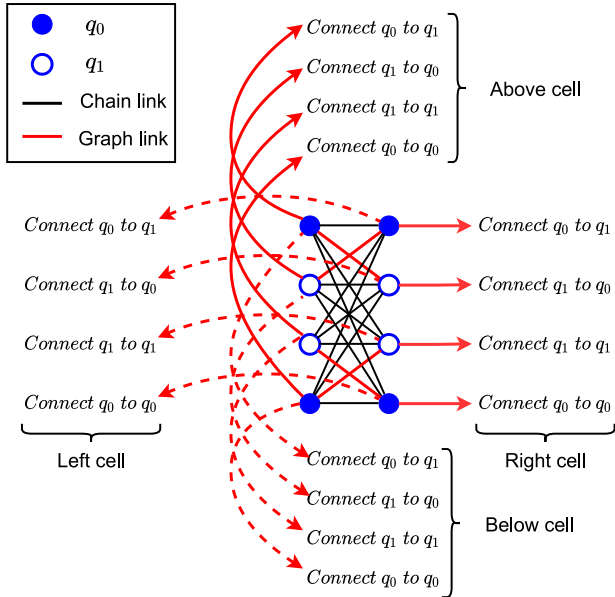
The embedding process varies based on the problem and the architecture of a given quantum annealer. D-wave systems provides a tool [1] that heuristically embeds the binary variables to the quantum annealer’s qubits. In our experience, the tool does not provide the optimal embedding in terms of keeping more related qubits closer to each other. Hence, we used a manual approach to embed the logical binary variables onto the physical qubits.

In manually embedding our variables onto qubits, we strived to produce a symmetric embedding. The symmetry of the embedding contributed to improved solution quality.

To each pixel, there corresponds a label. This is the unknown integer variable, which we are trying to determine its value by minimizing the cost function developed earlier. Assuming the values the labels attain are always less or equal to a maximum value of  $D$ , then one needs  $d = \lceil \log_2 D \rceil$  binary bits to encode the label (integer variable). In an ideal case, where the underlying topology of the quantum annealer is a complete graph, one would map each of the binary variables of the encoding to a single qubit. In cases where the underlying topology is not a complete graph (as in the case of the D-wave annealers), one needs to map a binary variable to several linked qubits to ensure the availability of a sufficient number of couplers to express the coefficients of the cost function. The chosen qubits are chain-linked together, i.e., have the linking couplers set to a maximum value to ensure that all the linked qubits attain the same value when they are read.

By expanding (12), we can observe that each binary variable  $x_i$  (logical bit) needs to be connected (because it falls in the same term) to all binary variables belonging to the same integer variable (the same pixel) and all the binary variables belonging to the connected integer variables (the connected pixels). In our implementation, each pixel is connected to its four adjacent pixels. This is the minimum number of connections needed to unwrap an image using a grid that connects all the pixels horizontally and vertically. This means that each binary variable needs to be connected to  $5d$  binary variables. First, it needs to be connected to  $d$  binary variables belonging to the same pixels. Second, it needs to be connected to  $4d$  binary variables belonging to the four neighboring pixels.

In addition to the couplers between the qubits in the same cell, each qubit of a Chimera cell is connected to two qubits in an adjacent cell as discussed earlier. Therefore, to allow a four-neighbor connectivity one needs to chain at least two qubits. This will realize an embedding, where  $D = 2$ , i.e., a problem with a maximum label value of 2. For more complex problems, one would opt of chain linking more than two qubits. In this work, we have elected to experiment with problems with a maximum label of 4. This is a realistic constraint as the real images in our dataset have labels with a maximum value of 5. In the remaining of the section, we shall discuss our approach to embedding the superpixel approach on the



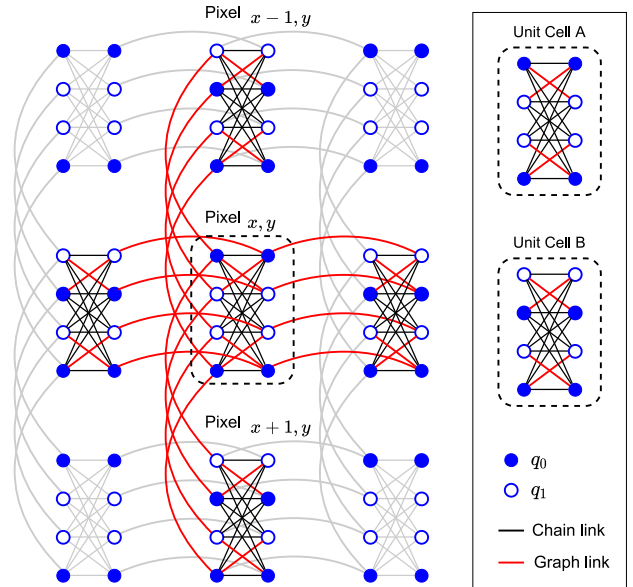
**FIGURE 3.** Embedding of an integer variable comprised of two binary variables ( $q_0$  and  $q_1$ ) into a unit cell and the connections with the adjacent cells.

D-wave 2000Q\_6 system. We will use two binary variables  $q_0$  and  $q_1$  to encode a maximum-label of 4 problem. We shall map each binary variable to four chain-linked qubits. Choosing four qubits to chain-link allows for a graph of degree 8, which is sufficient to express the four nearest neighbor connectivity of our problem.

Fig. 3 depicts the embedding of an integer variable comprised of two binary variables ( $q_0$  and  $q_1$ ) into a unit cell and the connections with the adjacent cells. In the figure, the chain link (in black) represents the chains between the qubit within the cell (the chain that constructs the integer variable), and the graph link (in red) represents the chains between qubits of different cells (the connections between integer variables).

Since there are no couplers available in the Chimera graph between the qubits within the cell column, the four qubits needed to be chained (to connect to the adjacent cells) must not all fall within the same column of the cell. Otherwise, we will not be able to chain them. Therefore, we selected the first and the last rows of the unit cell to represent the least significant bit of the integer variable ( $q_0$ ) and the middle rows to represent the most significant bit ( $q_1$ ) and called this representation unit cell A. Moreover, the last two rows' representation must be switched for the cells adjacent to cell unit A to allow different bits of different integer variables to connect (we call this representation unit cell B).

To sum up, we mapped each integer label to a single Chimera cell. In the present embedding, each integer label represents four distinct label values and is encoded using two binary variables. With each binary label variable mapped to 4 qubits of the D-wave annealer, the integer label is mapped to a single Chimera cell (a Chimera cell comprises eight qubits).



**FIGURE 4.** Manual embedding mapping.

The interconnections between the qubits of the same logical bit (chain links), the interconnections between the logical bits within the cell, and the interconnections between the cells (graph links) are shown in Fig. 4.

The largest image our manual embedding approach can map onto the D-wave 2000Q\_6 is  $16 \times 16$  pixels in size. Bypassing the faulty qubits resulted in asymmetries, which lowered the quality of the solutions. To avoid such issues, we have experimented with smaller images of  $10 \times 10$  pixels in size which, when mapped, avoid the faulty qubits with a concomitant increase in the quality of the solution.

In order to translate the state of qubits to the problem binary variables, the obtained solution from the annealer must be unembedded. That is translating the values of the chained qubits into a binary value. Since the chained qubits represent the same binary variable, their values must agree. If the values are different due to the probabilistic nature of the quantum annealer, we use the majority vote to determine the value of the binary variable.

In experimenting with the two options, i.e., the manual embedding and the automatically generated embedding, we found that on average, the automatically generated embedding provided unwrapping solutions with a matching fraction of 62%, while the manual embedding resulted in an average matching fraction of 83%. We obtained these results through the use of Ocean SDK software, provided by D-wave. We used ten synthetic images of size  $10 \times 10$  pixels. We used  $10 \times 10$  images because the automatic embedding mostly fails in finding a mapping for images that are larger than  $10 \times 10$  pixels.

### III. METHODOLOGY

In this section, we describe our methodology in breaking down the phase-unwrapping problem into smaller problems

that are easier to solve. The smaller problems can be solved in parallel, and approach the global solution.

**A. SINGLE-PASS SUPERPIXEL DECOMPOSITION**

Equation (7) describes the energy function for phase unwrapping. The equation consists of two terms, i.e., the pairwise term  $W_{st}(k_t - k_s - a_{st})^2$  that describes the relationship between two pixels  $s$  and  $t$ , and the unary term,  $\omega_s(k_s - a_s)^2$ , that describes the energy of the pixel  $s$ .

Single-pass superpixel decomposition is based on a divide-and-conquer approach. A given large InSAR image is subdivided into smaller subimages that fit onto a quantum annealing machine. Each of these subimages are unwrapped independently. Then the subimages are stitched together to form the final unwrapped large InSAR image.

This approach will introduce a problem at the boundaries of the subimages. Since we are unwrapping the subimages separately, there is no guarantee that the labeling of two pixels, each belonging to a boundary of two adjacent subimages, will attain optimal labeling consistent with that which would be obtained if both pixels had been part of the same optimization problem (i.e., if we had unwrapped the entire image all at once).

We assume that the pixels in each subimage are labeled correctly up to an integer additive factor, where all the pixels of one subimage share the same additive factor. Then, our objective is to find those additive factors such that all the pixels at the boundaries will be consistent.

We propose a superpixel heuristic to determine these additive factors as follows.

The wrapped InSAR image is divided into nonoverlapping subimages, where each subimage contains a subset of the pixels. The energy function determined by (7) can be rewritten as

$$E = \sum_{g \in G} \left[ \sum_{(s,t) \in A_g} W_{st} (k_t - k_s - a_{st})^2 + \sum_{s \in A_g} \omega_s (k_s - a_s)^2 \right] + \sum_{\substack{t \in A_i, \\ s \in A_j, \\ i \neq j}} W_{st} (k_t - k_s - a_{st})^2 \quad (14)$$

where  $G$  is the set of the subimages, and  $A_x$  is the set of pixels in the subimage  $x$ .

In the equation above, the terms within the square brackets correspond to an energy function for each subimage, while the last sum collects all the terms that connect the subimages. Our approach is to optimize each of the subimages, that is, to determine the labels that optimize the energy functions corresponding to each subimage separately. We assume next that the obtained solutions are correct—that are identical plus or minus a subimage wide integer shift—to the solution obtained when we optimize the image in its totality. The next step is to determine these additive factors, which can be formulated as a QUBO problem. Let  $K_s$  denote additive factor corresponding to subimage  $s$ , and let  $k'_i$  denote the label

of pixel  $i$  as determined by the QUBO of each subimage. Then label  $k_i$  of pixel  $i$  can be written as  $k_i = k'_i + K_s$  for  $i$  in  $A_s$ .

Equation (14) can now be rewritten as

$$E = \sum_{g \in G} \left[ \sum_{(s,t) \in A_g} W_{st} (k'_t + K_g - k'_s - K_g)^2 + \sum_{s \in A_g} \omega_s (k'_s + K_g - a_s)^2 \right] + \sum_{\substack{t \in A_i, \\ s \in A_j, \\ i \neq j}} W_{st} (k'_t + K_i - k'_s - K_j - a_{st})^2. \quad (15)$$

The first sum within the bracket is devoid of  $K_g$  and it is constant since the labels  $k'_s$  have been determined by the previous QUBO operation. Ignoring constant terms, (15) can be rewritten as

$$\tilde{E} = \sum_{g \in G} \left[ \sum_{s \in A_g} \omega_s (k'_s + K_g - a_s)^2 \right] + \sum_{\substack{t \in A_i, \\ s \in A_j, \\ i \neq j}} W_{st} ((k'_t - k'_s) + (K_i - K_j) - a_{st})^2 \quad (16)$$

denoting as  $a'_s = a_s - k'_s$  and  $a'_{st} = a_{st} - (k'_t - k'_s)$ , then (16) is written as

$$\tilde{E} = \sum_{g \in G} \left[ \sum_{s \in A_g} \omega_s (K_g - a'_s)^2 \right] + \sum_{\substack{t \in A_i, \\ s \in A_j, \\ i \neq j}} W_{st} (K_i - K_j - a'_{st})^2. \quad (17)$$

The first term of (17) is a second-order function, while the second term regularizes the solution by selecting  $K_g$  to be as small as possible. The coefficient  $\omega_s$  and the term  $a_s$  as per (7) and (8) were chosen arbitrarily. To ensure that our energy function conforms to the form of (7) and without affecting the accuracy of the solution, we select  $\omega_s = \omega_g$  and  $a'_s = a_g \forall s \in A_g$ . This results in the following expression for the energy function:

$$\hat{E} = \sum_{g \in G} W_{st} (K_i - K_j - a'_{st})^2 + \sum_{g \in G} \omega_g (K_g - a_g)^2. \quad (18)$$

This is the energy function of the superpixel level, where  $K_g$  represents the sought labels for each subimage (i.e., superpixel). Since it is a quadratic unconstrained integer optimization problem, it is amenable to a QUBO solution.

### B. MULTIPASS SUPERPIXEL DECOMPOSITION

In the single-pass superpixel decomposition explained above. The image is unwrapped in two stages. First, unwrap the subimages, and then unwrap the superpixel image. The incentive was to fit the problem onto a quantum annealing machine. The size of the subimage is limited by the size of the problem the quantum annealer can solve. The size of the superpixel image is in turn determined by the subimage size. Since the superpixel image is also unwrapped by a quantum annealing machine, the superpixel image might not fit onto the quantum annealing machine. In this case, the super-pixel can be divided into subimages and the super-pixel algorithm can be performed again to obtain a solution for the super-pixel image. In other words, the image will be unwrapped using two passes of the super-pixel algorithm.

In this way, the super-pixel algorithm can be recursively applied through multiple passes of the super-pixel decomposition.

For instance, consider an image of size  $400 \times 400$  pixels, while the annealer can handle at maximum  $10 \times 10$  pixels. Selecting the subimage size of  $10 \times 10$  will result in a super-pixel image of  $20 \times 20$  pixels, where each pixel represents a subimage additive integer. The  $20 \times 20$  super-pixel image will not fit into the annealer. Hence, we consider reapplying the super-pixel algorithm again on the super-pixel image by dividing it into four images of the size  $10 \times 10$  pixels. This will result in a  $2 \times 2$  second pass super-pixel image to unwrap the  $20 \times 20$  first pass super-pixel image.

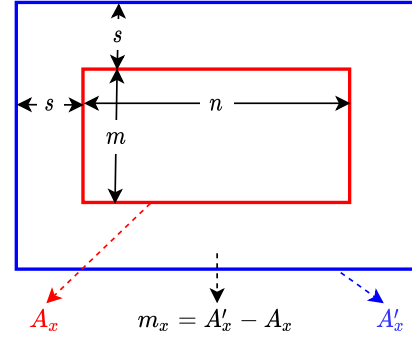
### C. ADDING A MARGIN TO THE SUBIMAGES

We postulated earlier that the pixels in each of the subimages were labeled correctly (within an additive constant) when these subimages were unwrapped independently of each other. However the boundary pixels belonging to neighboring subimages do play a role in deciding the label of the pixels in the subimage in question. Thus, labeling errors are introduced. In order to improve the quality of the unwrapping, we propose to unwrap subimages that overlap. That is, include a margin of  $s$  pixels around the subimage. However, as we proceed with the second phase of our method, we only keep the labels of the nonoverlapping parts of the subimage. The approach is shown in Fig. 5, and discussed in the subsequent paragraphs.

Consistent with the notation introduced above, we denote by  $A'_x$  the set of pixels of the overlapping subimage  $x$ , while the set of pixels of the nonoverlapping subimage is denoted as  $A_x$ . The set of pixels in the margin is, therefore,  $m_x = A'_x - A_x$  where the minus ( $-$ ) sign denotes a set-theoretic subtraction.

Equation (14) can now be written as

$$E = \sum_{g \in G} \left[ \sum_{(s,t) \in A_g} W_{st} (k_t - k_s - a_{st})^2 + \sum_{s \in A_g} \omega_s (k_s - a_s)^2 + \sum_{(s,t) \in m_g} W_{st} (k_t - k_s - a_{st})^2 \right]$$



**FIGURE 5.** Approach of adding margin to the subimage. The red box represent the subimage with  $A_x = m \times n$  pixels, the blue box is the subimage after adding the margin with total  $A'_x = (m + s) \times (n + s)$  pixels, and the area in between is the margin of  $m_x = A'_x - A_x$  pixels added around the subimage.

$$\begin{aligned} & + \sum_{s \in m_g} \omega_s (k_s - a_s)^2 + \sum_{\substack{s \in m_g, \\ t \in A_g}} W_{st} (k_t - k_s - a_{st})^2 \\ & + \sum_{\substack{s \in A_g, \\ t \in m_g}} W_{st} (k_t - k_s - a_{st})^2 - \sum_{(s,t) \in m_g} W_{st} (k_t - k_s - a_{st})^2 \\ & - \sum_{s \in m_g} \omega_s (k_s - a_s)^2 - \sum_{\substack{s \in m_g, \\ t \in A_g}} W_{st} (k_t - k_s - a_{st})^2 \\ & - \sum_{\substack{s \in A_g, \\ t \in m_g}} W_{st} (k_t - k_s - a_{st})^2 \Big] \\ & + \sum_{\substack{t \in A_i, \\ s \in A_j, \\ i \neq j}} W_{st} (k_t - k_s - a_{st})^2. \end{aligned} \quad (19)$$

In (19) we added and subtracted the terms that correspond to the margins of the subimages. We can now consolidate the margins with the nonoverlapping subimages to formulate an energy function based on the overlapping subimages as follows:

$$E = \sum_{g \in G} \left[ \sum_{(s,t) \in A'_g} W_{st} (k_t - k_s - a_{st})^2 + \sum_{s \in A'_g} \omega_s (k_s - a_s)^2 - \sum_{(s,t) \in m_g} W_{st} (k_t - k_s - a_{st})^2 - \sum_{s \in m_g} \omega_s (k_s - a_s)^2 - \sum_{\substack{s \in m_g, \\ t \in A_g}} W_{st} (k_t - k_s - a_{st})^2 - \sum_{\substack{s \in A_g, \\ t \in m_g}} W_{st} (k_t - k_s - a_{st})^2 \right]$$



**TABLE 1. Properties of the Original Real Images**

Image Name	Max label	Size	Generated Image Size	# Generated Images
Mexico City	4	500 × 500	400 × 400	4
Las Vegas	4	960 × 1080	400 × 400	1
Vancouver	5	1619 × 974	400 × 400	4

$$+ \sum_{\substack{t \in A_i, \\ s \in A_j, \\ i \neq j}} W_{st} (k_t - k_s - a_{st})^2. \quad (20)$$

The energy function described by the above equation includes the energy function of the overlapping subimages, i.e.,  $E'(k) = \sum_{g \in G} [\sum_{(s,t) \in A'_g} W_{st} (k_t - k_s - a_{st})^2 + \sum_{s \in A'_g} \omega_s (k_s - a_s)^2]$ . We propose to optimize these overlapping subimage energy functions, i.e.,  $\arg \max_k E'(k)$  separately. Then, by introducing these labels back into (15), and following the subsequent steps outlined in Section III-A we can unwrap the entire image based on the labels computed through the margin enhanced subimages.

**IV. EXPERIMENTS**

In this part of the work, we present the experimental approach we followed in order to study the effectiveness of the proposed super-pixel method. Our methodology included the use of different solvers, a number of image datasets, and the analysis of the results we obtained. The datasets we used included synthetic SAR images as well as real InSAR images. We have used synthetic images as they afforded us the opportunity to control the complexity and their noise content. The solvers we employed included TRWS, which is the current industry standard as well as QUBO solvers including a quantum annealer (D-wave 2000Q\_6).

The TRWS solver is readily accessible and since it is the industry standard, it was used to study the effectiveness and establish the performance baseline of our proposed super-pixel method on a wide variety of images. The QUBO solvers being less readily accessible, were used in more restrictive experiments to demonstrate the feasibility of our proposed method on such solvers, and also compare their relative effectiveness.

**A. IMAGE DATASETS**

**1) REAL IMAGES**

A real InSAR images dataset is constructed using three large real InSAR images. The images are partitioned at random to have the size 400 × 400 to be consistent with the results of the synthetic images. The partitioning resulted in generating multiple images, in some case, when the original image size is large enough to fit more than one 400 × 400 image. The images have a maximum label that varies between 3 and 5. Table 1 summarizes the original images properties.

Table 2 summarizes the properties of the generated dataset.

**TABLE 2. Maximum Label for the Real InSAR Images Dataset**

Image Name	Max label
Mexico City	4
Las Vegas 0	4
Las Vegas 1	3
Las Vegas 2	4
Las Vegas 3	3
Vancouver 0	5
Vancouver 1	5
Vancouver 2	5
Vancouver 3	5

**2) SYNTHETIC IMAGES DATASET 1**

This is the main synthetic dataset. This dataset presents a larger exploration space, with a wide spectrum that includes high-frequency data that present a challenge for the phase unwrapping process. 400 × 400 pixels synthetic images are generated with the following parameters.

- 1) Image complexity (three levels low, medium, and high).
- 2) Noise content (four levels free, low, medium, and high).
- 3) The number of labels (three different numbers of labels; 4, 8, and 16).

This will give us 36 different sets. Ten images are generated per set to give in total 360 images. The synthetic images are generated using Perlin noise generator. Table 3 summarizes the properties of the generated images.

**3) SYNTJEYIC IMAGES DATASET 2 (PSEUDOREAL)**

This dataset includes images that have a spectrum that is similar to that of the real images’ dataset. We postulate that such synthetic images have computational complexity similar to that of the real images and our methodology can be evaluated fairly using these synthetic images. We generated ten images with 400 × 400 pixels. The maximum label (maximum number of wrappings) of the real images varies between 3 and 5 with average maximum label of 4 (Table 3). Therefore, we generated the pseudo-real images with the maximum label of 4. The images are considered of low complexity with a Perlin noise correlation of 0.5 and very noisy with an SNR of 3 db.

**B. METRICS**

To determine how close two images (of identical size) are to each other, we use the matching fraction metric defined as the fraction of pixels that are identical in the two images. There are two ways to characterize the efficacy of a new method. The first is to compare its efficacy to that of an oracle. The second is to compare its efficacy to that of an established method, which in many cases is considered to be the state of the art. Specifically for our case, we shall follow the second approach and compare the efficacy of our method to that of the whole image processed using TRWS. Since both our method and the whole image TRWS introduce

**TABLE 3. Properties of the Images Generated for the Synthetic Dataset 1**

Max label	Low- noise SNR (dB)	Medium- noise SNR (dB)	High- noise SNR (dB)	Low- complexity correlation (Perlin noise)	Medium- complexity correlation (Perlin noise)	High- complexity correlation (Perlin noise)
4	13	10	7	12	18	24
8	13	10	7	6	9	12
16	13	10	7	3	4.5	6

uncorrelated errors as compared to the oracle, comparing directly the resulting images would include both sets of errors. Rather, our approach is to derive the efficacy of the state-of-the-art whole-image-TRWS in reference to the oracle (i.e., the ground truth) as well as the efficacy of our method in reference to the same ground truth and then compare the two efficacies. The following algorithm outlined the method we just discussed.

For each wrapped image:

- 1) establish the corresponding ground truth (that is the unwrapped image an oracle would derive);
- 2) unwrap the wrapped image using the whole-image-TRWS method
  - a) obtain the matching fraction of the unwrapped image to the ground truth and name it  $mf_{TRWS}$ ;
- 3) unwrap the wrapped image using the proposed method
  - a) obtain the matching fraction of the unwrapped image to the ground truth and name it  $mf_{method}$ ;
- 4) compare the matching fractions through the effectiveness index defined as

$$R = \frac{mf_{method}}{mf_{TRWS}}$$

Depending on the image chosen different ground truths may be used. For example, SAR satellite images are not accompanied by the corresponding unwrapped, noise-free image. Rather, a state-of-the-art method is used to unwrap such images, and the resulting images are deemed as the ground truth. On the other hand, the process of generating synthetic images results in both the noise-free unwrapped image and its corresponding noise-free or noisy wrapped one. Since we have experimented with both satellite and synthetic images, different ground truth images are used in the various experiments. The terminology below defines the possible cases. To ensure fairness, we use the same ground truth image to compare the efficacy of our method to the state-of-the-art whole-image TRWS. Following, we enumerate the possible “ground truth” cases we used in this research.

#### 1) NOISE-FREE GROUND TRUTH

This is the original image obtained through an ideal noise-free sensor. However, such images cannot be had in reality as sensors introduce noise and artefacts. Synthetic images though, can be synthesized as noise and artefact free and they can be used as noise-free ground truth images in our experimentation and analysis.

#### 2) NOISY UNWRAPPED GROUND TRUTH

Images are the images obtained by a sensor. Such images are common in photography; however, SAR (and InSAR) images are wrapped. Noisy unwrapped ground truth images can be obtained synthetically by adding noise to synthetic noise-free ground truth images.

#### 3) DEEMED NOISY UNWRAPPED GROUND TRUTH

Images are images that were obtained from wrapped noisy images through the application of a state-of-the-art unwrapping method. For our purposes, the state-of-the-art unwrapping method used is TRWS.

In this work, we will primarily use the noisy unwrapped ground truth images as the base to compare the images obtained as a result of our methods, where the noisy unwrapped ground truth does not exist, we will use the deemed noisy unwrapped ground truth. These are the cases of real images.

### C. SOLVERS

#### 1) TRWS

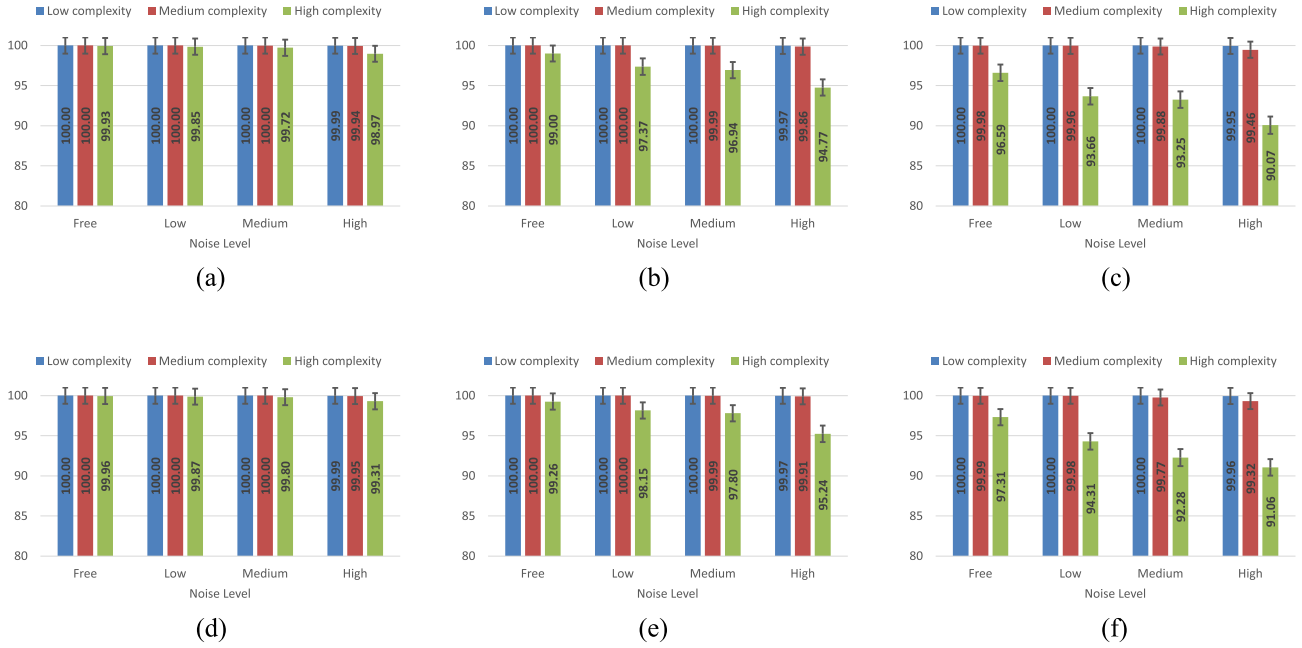
Since we have no restrictions on using TRWS, we will use it to provide a detailed analysis of the performance of our super-pixel approach in solving the unwrapping problem. The experiments using TRWS will be applied to all images in the synthetic dataset1 and/or the real and the pseudo-real datasets.

#### 2) QUBO

QUBO solvers extend the solvers used with the super-pixel algorithm to include a hardware-based QUBO solver (D-wave annealer) and software-based QUBO solvers (MQIO solvers). The discussion presented in the TRWS experiments provides a detailed analysis of the performance of our super-pixel approach in solving the unwrapping problem.

Since the quantum solvers are not widely accessible and cannot yet support very large problems, by necessity our experiments are limited in the size and the number of problems we can solve. The experiments here are to be considered as complementary and validating the analysis we presented in the previous sections.

Given the limited access we have to the QUBO solvers, we had to choose between applying the one-pass super-pixel approach or the two-pass super-pixel approach. We used the two-pass super-pixel approach as it suits our experiments with the QUBO solvers best (large images of  $120 \times 120$  pixels with small subimages of maximum size of  $10 \times 10$  pixels). Also, as we shall see, the two-pass super-pixel algorithm



**FIGURE 6.** Geometric mean of the effectiveness index of the images in dataset 1 obtained through the one-pass super-pixel algorithm for different maximum labels, and different subimage sizes compared to the noisy unwrapped ground truth. (a) Max label 4, sub-image  $20 \times 20$ . (b) Max label 8, sub-image  $20 \times 20$ . (c) Max label 16, sub-image  $20 \times 20$ . (d) Max label 4, sub-image  $30 \times 30$ . (e) Max label 8, sub-image  $30 \times 30$ . (f) Max label 16, sub-image  $30 \times 30$ .

**TABLE 4.** Configurations of the D-Wave Annealer

Property	Value	Property	Value
chain scale	0.6	use_ising	True
num_reads	100	manual_embedding	True
chain_strength	1	solver_name	'DW_2000Q_6'
h_threshold	0.0003	max_label	4
J_threshold	0.002	encoding	binary
auto_scale	False		

underperforms the one-pass super-pixel one. Hence, it is expected the one-pass super-pixel algorithm to yield superior performance should the solver have sufficient resources (e.g., number of qubits) to accommodate it.

For all the images we used with the QUBO solvers, the images are cropped to the size of  $120 \times 120$ . We selected the cropped part randomly within the selected image. The selected size of the image fits different subimage sizes. The subimage sizes we used are  $6 \times 6$ ,  $8 \times 8$ , and  $10 \times 10$ .

- 1) *D-Wave Annealer*: The super-pixel algorithm is tested in this work using a D-wave annealer (D-wave 2000Q\_6 with 2000 qubits) as a subimage solver. The D-wave annealer is configured to maximum label = 4. Table 4 summarizes the configurations for the D-wave annealer.
- 2) *MQIO Solvers*: We have used Microsoft’s quantum-inspired optimization (QIO) as a subimage solver for our super-pixel algorithm. The two solvers employed were simulated annealing and parallel tempering. This experiment provided us with a mean of comparison

to the efficacy of the D-wave solver since all three implement similar optimization approaches.

## V. EVALUATING THE SUPER-PIXEL ALGORITHM

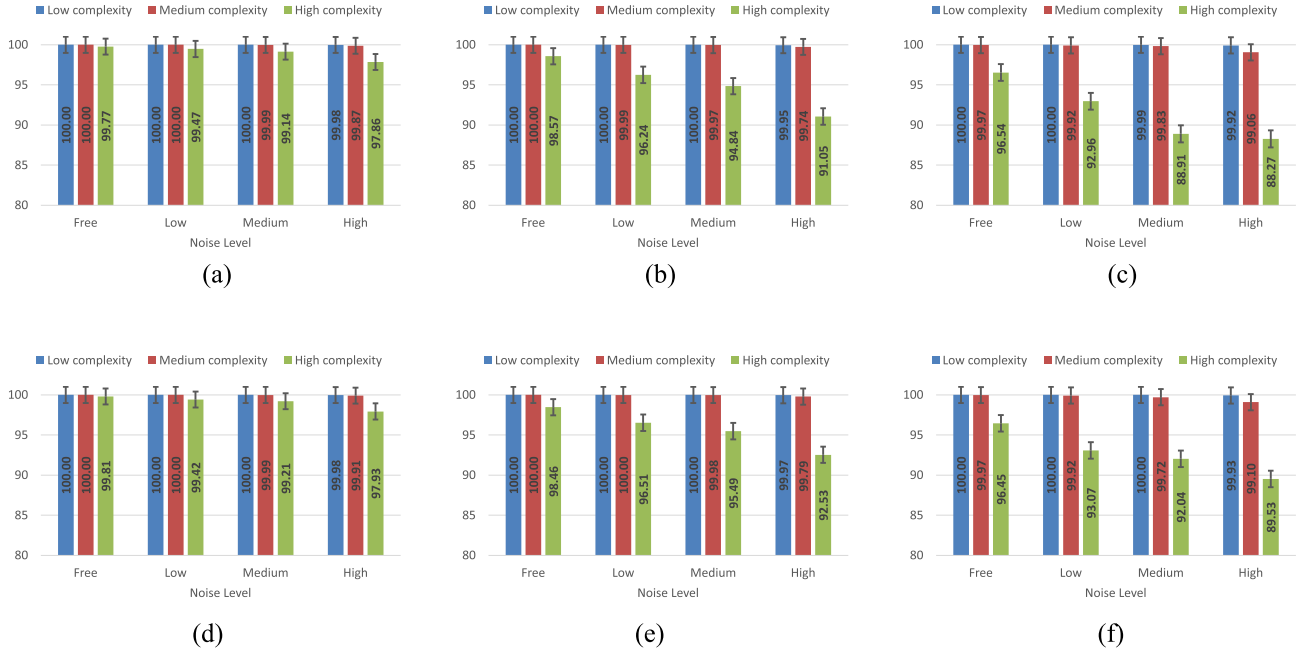
In this section, we test the one-pass super-pixel and the multipass super-pixel algorithms. We first use TRWS as the subimage solver and subsequently, we use the different solvers we discussed earlier.

### A. TRWS AS SUBIMAGE SOLVER

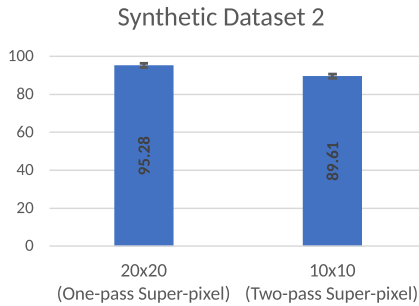
In this subsection, we evaluate the one-pass super-pixel and the two-pass super-pixel algorithms using TRWS as subimage solver. For both cases, the original image is partitioned into subimages, each subimage is processed separately and then the results are combined following the one-pass or the multipass superpixel algorithm as discussed in Sections III-A and III-B. The sizes of the subimages were adjusted to accommodate the requirements of the method (single or multipass) used.

#### 1) SYNTHETIC DATASET 1 RESULTS

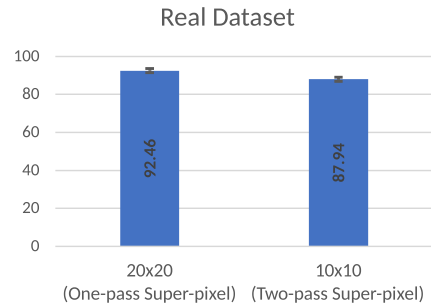
We experimented with all the synthetic images in dataset 1 (360 images in total). All images are  $400 \times 400$  pixels. We tried the one-pass super-pixel algorithm (using  $20 \times 20$  and  $30 \times 30$  subimages) and the two-pass algorithm (using  $10 \times 10$  and  $14 \times 14$  subimages). We compare the images obtained by our method to the images obtained by applying TRWS on the entire unwrapped image using effectiveness index as per the discussion in Section IV-B.



**FIGURE 7.** Geometric mean of the effectiveness index of the images in dataset 1 obtained through the two-pass super-pixel algorithm for different maximum labels, and different subimage sizes compared to the noisy unwrapped ground truth. (a) Max label 4, sub-image 10 × 10. (b) Max label 8, sub-image 10 × 10. (c) Max label 16, sub-image 10 × 10. (d) Max label 4, sub-image 14 × 14. (e) Max label 8, sub-image 14 × 14. (f) Max label 16, sub-image 14 × 14.



**FIGURE 8.** Geometric mean of the effectiveness index of the images in dataset 2, compared to the deemed noisy unwrapped ground truth. The error bars depict the standard deviation.



**FIGURE 9.** Geometric mean of the effectiveness index of the images in the real dataset, compared to the deemed noisy unwrapped ground truth. The error bars depict the standard deviation.

Figs. 6 and 7 present the efficacy of the one-pass and two-pass super-pixel algorithm as compared to the state-of-the-art TRWS method applied on the entire unwrapped image. The ground truth chosen for this experiment was the noisy unwrapped ground truth.

Analyzing the results as presented in Figs. 6 and 7, one can draw some general observations. The noise and image complexity are the important factors in the quality of the unwrapping process.

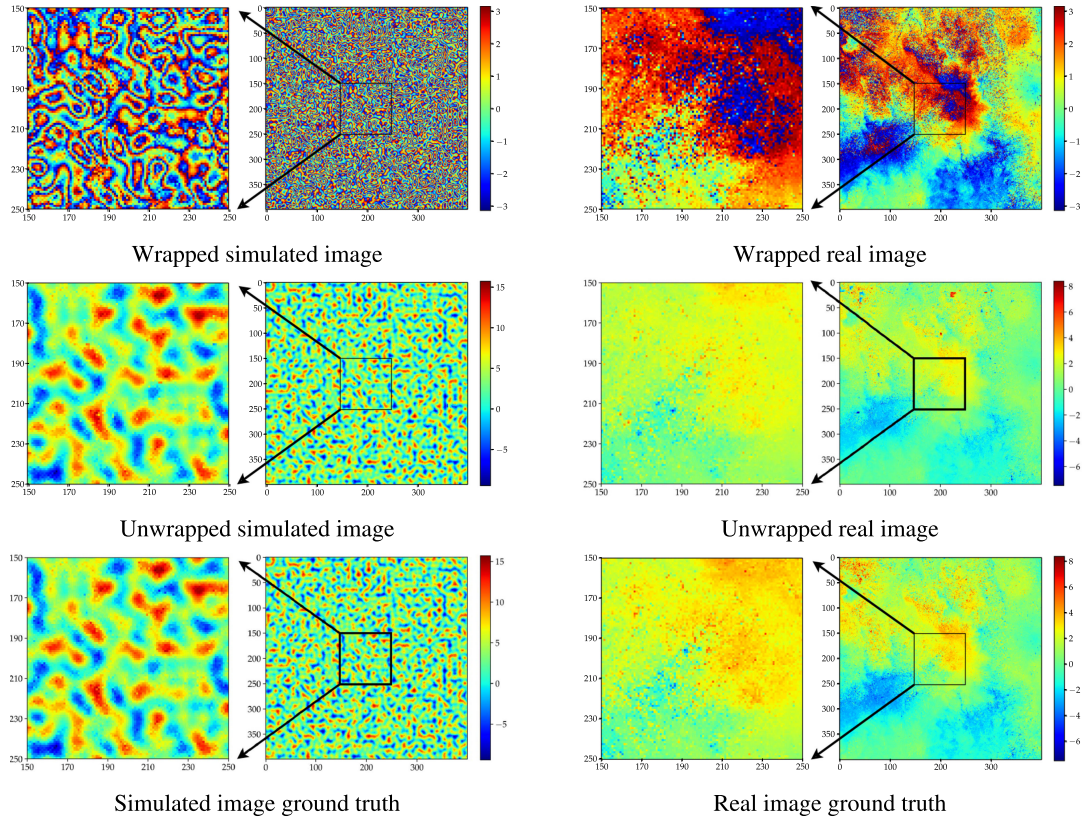
If the images are noise-free and of low or medium complexity (blue and red bars in the figures), our approach unwraps the image essentially identically to the entire-image-TRWS method (the effectiveness index is larger than 0.99) and produces identical results with the ground truth. This can be seen in Figs. 6 and 7. However, as these figures show, the complexity of the image affects the quality of the solution

of our approach marginally. Our approach, in the worst case of highly complex images and a large number of labels (16), deteriorates by 3.4% to an effectiveness index of 0.966 as shown in Fig. 7(c).

As one would expect, noise plays a major role in the success of unwrapping algorithms. In the worst case, depicted in Fig. 7(c), the quality of the resulting image as compared to the original noisy image deteriorates by about 11.7% to an effectiveness index of 0.8827. This result covers images with high noise and high complexity.

As the size of the subimages decreases, there is some deterioration of the quality of the solution, which is more pronounced for high-complexity and high-noise images. In the worst case, the deterioration is about 1.25% [0.8953 for a high noise high complexity 16-label image to 0.8827 Figs. 6(f) and 7(c)].





**FIGURE 10.** Samples from the images unwrapped using the super-pixel approach. The left-hand column shows the unwrapping of a simulated high-noise image, while the right-hand column shows the unwrapping of a real image. The color bars on the right side of the figures represent the phase value in radian. On the top row the wrapped images, ranging in value from  $-\pi$  to  $\pi$ . In the middle row the unwrapped images, ranging in value from  $-\pi$  to  $\pi$ , unwrapped using the super-pixel algorithm when TRWS is used as subimage solver. At the bottom row the ground truth images to compare against, ranging in value from  $-\pi$  to  $\pi$ .

Overall, we can conclude that both the one-pass and the two-pass super-pixel algorithms produce excellent results for low- and medium-complexity images [the worst case is depicted in Fig. 7(c) with an effectiveness index of 0.9906]. Real SAR and InSAR images are typically of low or medium complexity. Our approach produces acceptable results for high-complexity images [the worst case is depicted in Fig. 7(c) at 0.8827].

## 2) REAL IMAGE AND PSEUDO-REAL IMAGE DATASETS RESULTS

This section presents and compares the performance of the super-pixel algorithm on the synthetic dataset 2 (pseudo-real) and the real dataset.

We experimented with the synthetic images in dataset 2 (ten images in total) and the real images dataset (nine images in total). All images are  $400 \times 400$  pixels. We tried the one-pass super-pixel algorithm using  $20 \times 20$  subimages and the two-pass algorithm using  $10 \times 10$  subimages. In both cases, we used TRWS as the subimage solver and compared it to the deemed noisy unwrapped ground truth. We selected to compare to the deemed noisy unwrapped ground truth because the noisy unwrapped ground truth does not exist for the real dataset. As discussed in [11], the deemed noisy unwrapped

ground truth and the noisy unwrapped ground truth images are almost identical. We summarized the results in Figs. 8 and 9.

## 3) DISCUSSION

The results obtained with real images are comparable to those of the synthetic images (pseudo-real) in dataset 2. Both datasets produce results with an effectiveness index within 1% of each other (cf. Figs. 8 and 9).

The results obtained are congruent with the results obtained in Section V-A1 above. The images analyzed in this section have a much higher noise content (SNR of 3 db) as compared to the images in dataset 1 (SNR of 7 db for high noise images). As we discussed in Section V-A1, noise and complexity contribute to the rapid deterioration of the quality of the solution. Using a larger subimage gives better results.

It seems that the deterioration of the effectiveness of the super-pixel method for highly complex and very noisy images correlates with the violation of the Nyquist criterion introduced by noise [11]. If a subimage has a large number of aliased pixels, the lack of an extended field of unaliased pixels from the neighboring subimages prevents the optimizer from assigning the correct labels. The larger the subimage,

the larger the neighboring field of unaliased pixels and the better the chance of assigning the correct label.

#### 4) INSAR IMAGES SAMPLES DISCUSSION

Fig. 10 shows the results of the unwrapping process on two images. the first is a synthetic image, taken from synthetic dataset 1, while the second is a portion of a real image (Vancouver image).

In both cases, we present the wrapped image, the ground truth (which for the case of the real image is the deemed noisy ground truth), and the unwrapped image resulting from the application of our super-pixel method.

In both cases, especially for the synthetic image, one can verify the close proximity of the super-pixel-obtained image to the ground truth.

The differences observed are mostly due to noise, which alters the Nyquist assumption [11].

#### B. QUBO AS SUBIMAGE SOLVER

The primary focus of the experimentation is to study the applicability of QUBO solvers to our approach and to compare the relative performance of these solvers. Given the limitation in the accessibility of QUBO solvers and their limitations in accommodating very large problems, we have elected to experiment with a small number of reduced-size images. Specifically, we shall use subimages of maximum size  $10 \times 10$  and maximum label of 4. This is dictated by the size (in qubits) of the D-wave quantum annealer.

As we discussed in Section IV-C2, we shall employ the two-pass super-pixel algorithm on images of size  $10 \times 10$  and a maximum label of 4. However, not all the images in the different datasets are limited by a maximum label of 4. In this experiment set we will consider only the images that have a maximum label of 4 to study the applicability of QUBO solvers to the super-pixel algorithm.

In the experiments reported here, we gradually increase the subimage size to study its effect on the quality of the unwrapping. In the following experiments, we used one synthetic InSAR image and one real InSAR image to experiment with. We cropped the images at random to a size of  $120 \times 120$  pixels instead of the original large size ( $400 \times 400$ ). For this experiment set (and for the experiment in Section VI-B), we are experimenting with the same two images. We selected this maximum label so that the image could be fully accommodated by the various QUBO solvers we experimented with. We varied each experiment's configuration, i.e., the subimage size, the margin size, and the solver we used.

#### 1) RESULTS

We used our two-phase super-pixel algorithm on the two images selected as per the discussion above. We experimented using TRWS and QUBO solvers (D-wave, simulated annealing, and parallel tempering) as the subimage solvers. Our focus is the performance of the quantum annealer (D-wave 2000Q\_6). As such, we present the results of the TRWS

TABLE 5. Cases Considered for Experiment Set 1 (Testing a Real Image)

Case	Max label	Size	Sub-image size
0	4	$120 \times 120$	$6 \times 6$
1	4	$120 \times 120$	$8 \times 8$
2	4	$120 \times 120$	$10 \times 10$

TABLE 6. Cases Considered for Experiment Set 1 (Testing a Synthetic Image)

Case	Max label	Complexity	Noise	Size	Sub-image size
0	4	High	High	$120 \times 120$	$6 \times 6$
1	4	High	High	$120 \times 120$	$8 \times 8$
2	4	High	High	$120 \times 120$	$10 \times 10$

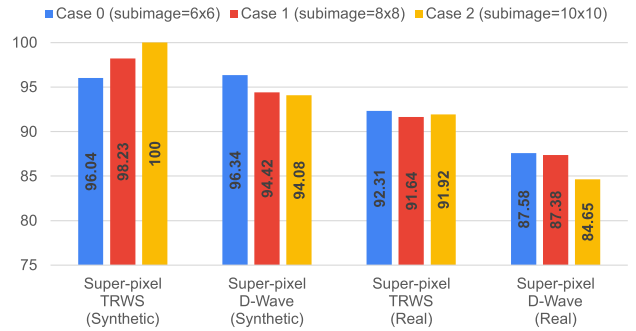


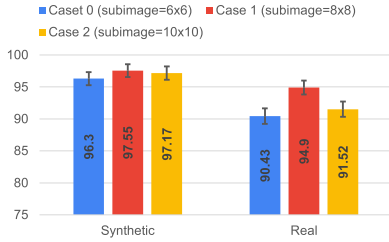
FIGURE 11. Super-pixel effectiveness index for a synthetic image and a real image for the different subimage sizes and different solvers (TRWS, D-wave).

and the D-wave solvers presently, while the results using the other QUBO solvers are presented in Appendix B.

We were forced to curtail our experimentation on a single image from each dataset due to the limitation in accessing the quantum annealer. To ensure that the chosen images were not biasing our results, we also processed two randomly selected subsets, of ten images from each of the sets of synthetic images 1 and real images. These images were processed using our two-pass super-pixel algorithm with TRWS as the subimage solver.

We assess the quality of the unwrapping as expressed by the effectiveness index. Since we had to select a single image from the synthetic dataset 1, we selected an image of high complexity and high noise. If the approach is giving good results for the highly complex and noisy image, it is more likely to give good results for images with lower complexity and less noise. Tables 5 and 6 show the cases we considered. All cases use the same image, and they are distinguished by the sizes of the subimages we have employed. We have used the geometric mean to calculate the average of the effectiveness index we report for the first set of results. The results are shown in Figs. 11 and 12.

As it can be verified by inspecting Fig. 11, the results of the chosen image are within one standard deviation of the mean of the results of all the images processed (Fig. 12).



**FIGURE 12.** Geometric mean of the effectiveness indexes of the randomly chosen  $120 \times 120$  pixel sections of images from the synthetic dataset 1 and from the real dataset. The error bars depict the standard deviation.

## 2) DISCUSSION

As presented earlier, using TRWS as a subimage solver for the super-pixel algorithm gives results that are close to the ground truth. In this experiment, the D-wave annealer, provide close results to the TRWS results when used as subimages solvers.

The one and the multipass super-pixel algorithm can be extended to use various phase unwrapping solvers; this includes the classical TRWS and the D-wave annealer. However, each subimage solver gives better results using specific configurations. Most of the QUBO solvers impose limitations on the number of pixels and the image’s size.

For D-wave, the subimage size  $6 \times 6$  has the best results for both; the average subimages matching and the super-pixel solution. This is because the D-wave annealer works better with smaller images (smaller images mean fewer qubits and better qubit coherence).

## VI. EVALUATION OF THE ADDITION OF MARGINS TO THE SUBIMAGES

In this section, we evaluate the impact of adding a margin to the subimages during the phases of the super-pixel algorithm. As we discussed in Section III-C, including a margin, provides additional constraints to the pixels at the boundaries of the subimages, and, thus, better informs the optimization process resulting in improved unwrapping quality. In the subsequent sections, we shall evaluate our approach first employing TRWS as the solver and then then a number of QUBO solvers.

### A. TRWS AS SUBIMAGE SOLVER

In this part of the work, we are analyzing the impact of introducing a margin around the subimages to the quality of the solution.

#### 1) ONE- AND TWO-PASS SUPER-PIXEL ALGORITHM

In this experiment, we kept the size of the subimage constant but gradually increased the size of the margin. The image size in dataset 1 are  $400 \times 400$ . Following the experimental design presented in Section V, we selected subimage sizes of  $20 \times 20$  for the one-pass and  $10 \times 10$  for the two-pass algorithm. The subimages were augmented with varying-sized margins and the super-pixel algorithm was used to unwrap

**TABLE 7.** Sizes of the Subimages and the Margins Used in Testing Adding Margin (One-Pass Super-Pixel)

Case A0	$20 \times 20$ margin of 0 & $20 \times 20$ no margin
Case A1	$20 \times 20$ margin of 1 & $22 \times 22$ no margin
Case A2	$20 \times 20$ margin of 2 & $24 \times 24$ no margin
Case A3	$20 \times 20$ margin of 3 & $26 \times 26$ no margin
Case A4	$20 \times 20$ margin of 4 & $28 \times 28$ no margin
Case A5	$20 \times 20$ margin of 5 & $30 \times 30$ no margin

**TABLE 8.** Sizes of the Subimages and the Margins Used in Testing Adding Margin (Two-Pass Super-Pixel)

Case B0	$10 \times 10$ margin of 0 & $10 \times 10$ no margin
Case B1	$10 \times 10$ margin of 1 & $12 \times 12$ no margin
Case B2	$10 \times 10$ margin of 2 & $14 \times 14$ no margin
Case B3	$10 \times 10$ margin of 3 & $16 \times 16$ no margin
Case B4	$10 \times 10$ margin of 4 & $18 \times 18$ no margin
Case B5	$10 \times 10$ margin of 5 & $20 \times 20$ no margin

the whole image. We also used the nonmargin super-pixel algorithm to unwrap the whole image, but with subimages having the same size as the ones that included the margins, and compared the two results. The experimental cases we considered are shown in Tables 7 and 8, while the results appear in Figs. 13 and 14. There, the solid lines correspond to the results of the margin-included super-pixel algorithm, while the dashed lines correspond to the results of the pure (no margin) super-pixel algorithm.

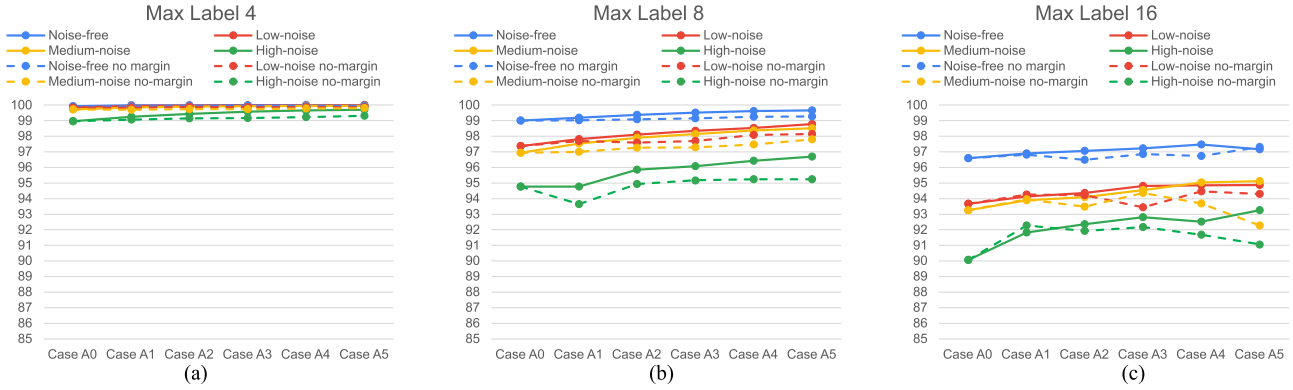
2) *Discussion:* As Figs. 13 and 14 demonstrate, adding a margin improves the results (the solid lines are above the dashed once). The ramification is that one would obtain a more accurate unwrapping should one elect to ignore the margins as compared to the case, where the margins are not ignored (the dashed line). Of course, the amount of computation involved increases as more subimages need to be processed in the case of the margin-included algorithm. However, since these subimages can be processed independently, and potentially in parallel, the overall computation time is not affected.

### B. QUBO AS SUBIMAGE SOLVER

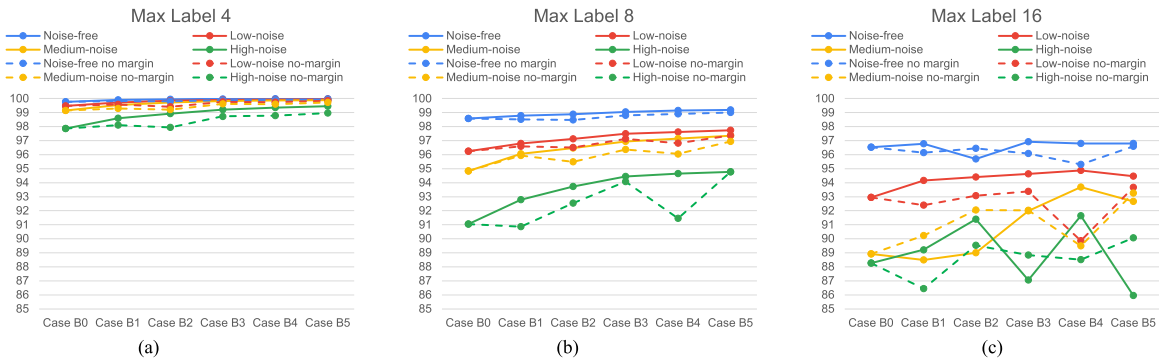
Having established the effectiveness index of adding a TRWS margin, we now study the effectiveness of the various QUBO solvers. Since we had limited access to the QUBO solvers, we have elected to use one of the images from our synthetic images database and one image from the real images dataset (the identical images selected previously for the experiments Section V-B1) and do a comparative analysis of the quality of the solutions achieved by the different QUBO solvers. For both cases, the selected images had a maximum label of 4 because of the limited problem size that the QUBO solvers, especially the D-wave 2000Q\_6, can accommodate.

Further, to reduce the number of computations required we randomly cropped the selected image to a size of  $120 \times 120$  pixels. The subimage sizes we used for our study are  $6 \times 6$ ,  $8 \times 8$ , and  $10 \times 10$ . We have also included the results of using TRWS as the solver to compare to the QUBO solvers’ results. However, since we have no restrictions on using TRWS, we experimented with all the images that have the





**FIGURE 13.** Bar graphs<sup>1</sup> of the matching fractions in terms of the margins employed. The subimage sizes and margins used are depicted in Table VII. These are the results of the one-pass margin-employing super-pixel method. (a) Max label 4. (b) Max label 8. (c) Max label 16.



**FIGURE 14.** Bar graphs<sup>1</sup> of the matching fractions in terms of the margins employed. The subimage sizes and margins used are depicted in Table VIII. These are the results of the two-pass margin-employing super-pixel method. (a) Max label 4. (b) Max label 8. (c) Max label 16.

same characteristics (same complexity, noise level, and maximum labels). To derive the matching fractions, we compare the results to the noisy unwrapped ground truth for the synthetic image and the deemed noisy unwrapped ground truth for the real image.

As per the analysis presented in Section VI-A, we would expect that in general adding a margin to the sub-images, would result in increased unwrapping quality.

As it will be shown, the use of QUBO solvers produces results that are comparable to those of produced using TRWS. In addition, adding a margin, in many cases improves the quality of the solution. However, the size of the subimages plays a significant role in the obtained quality.

1) CONSTANT TOTAL SIZE OF THE SUBIMAGE

Here, we increase the margin while keeping the total subimage size constant. Table 9 summarizes the properties of the images.

- a) *Synthetic image and real image results:* Tables 10 and 11 summarize the super-pixel matching fraction for the cases shown in Table 9. Table 10 shows the results for the synthetic image, while Table 11 shows the ones

<sup>1</sup>The separated bar graphs for each case can be found in the accompanied supplemental document.

**TABLE 9.** Cases Considered for Adding Margin While Fixing the Total Subimage Size With QUBO Solvers

Case	Max label	Size	Sub-image size	Margin	Sub-image total size
0	4	120 × 120	6 × 6	2	10 × 10
1	4	120 × 120	8 × 8	1	10 × 10
2	4	120 × 120	10 × 10	0	10 × 10

**TABLE 10.** Results of Adding Margin While Fixing the Total Subimage Size With QUBO Solvers and a Synthetic Image

Experiment	Sub-image	margin	Super-pixel	
			TRWS (Mean/STD)	D-Wave
Case 0	6 × 6	2	97.83/1.01	92.16
Case 1	8 × 8	1	98.15/1.00	92.07
Case 2	10 × 10	0	97.16/1.03	93.57

for the selected real image. In both tables, we show the results of TRWS being the subimage solver and compare them to those of the D-wave being solver. The results obtained when we used the MQIO solvers (simulated annealing and parallel tempering) are shown in Appendix C.

- b) *Discussion:* As shown in Tables 10 and 11, the results are congruent with the results of experimenting



**TABLE 11. Results of Adding Margin While Fixing the Total Subimage Size With QUBO Solvers and a Real Image**

Experiment	Sub-image	margin	Super-pixel	
			TRWS (Mean/STD)	D-Wave
Case 0	6 × 6	2	95.38/1.28	90.15
Case 1	8 × 8	1	93.85/1.30	90.84
Case 2	10 × 10	0	91.92/1.27	84.65

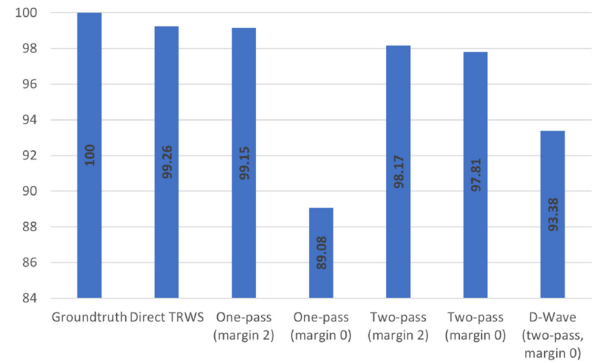
with the large synthetic dataset and presented in Section VI-A. That is, the inclusion of a margin improves the accuracy of the unwrapping. This can be seen for the TRWS experiment as we move from case 0 (large margins) through case 1 and case 2 (no margin) the performance decreases. The same holds mostly true for the D-wave results, especially for the real image. The results of the synthetic image are very close and one may attribute this variation on the statistical nature of the optimizer (D-wave) convergence. Similar behavior can be observed by the MQIO solvers (cf. Appendix C) highlighting again the statistical nature of the convergence of the optimizers.

However, the D-wave results are not statistically robust as these are based on a single image, while the TRWS results better represent the behavior of the method since they are representing the mean of processing ten different images.

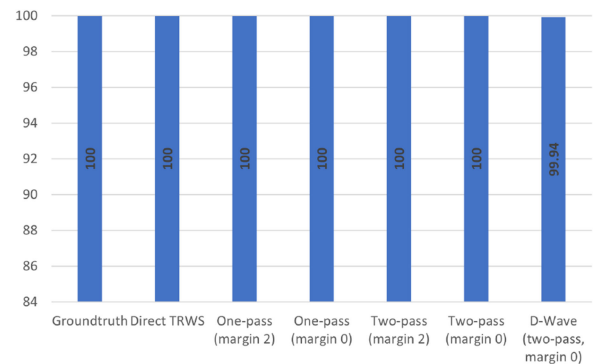
**VII. CONCLUSION**

In this work, we have successfully demonstrated that the SAR phase-unwrapping problem can be expressed and solved as a QUBO problem. By partitioning the problem, we have been able to obtain high-quality solutions for large images. We refer to this method as “single-pass super-pixel decomposition.” We extended the concept of the super-pixel decomposition to handle larger images by recursively partitioning the image until the partitions fit in the quantum annealer. Then recursively apply the second phase of the superposition until we obtain labels that approach the global solution (multipass super-pixel decomposition). We introduced the notion of a margin in processing subimages in our two-phase super-pixel unwrapping method. Margins enlarge the size of the subimages to be processed and result in overlapping subimages. The pixels in the margins provide additional information in the unwrapping process, and this results in increased accuracy of the resulting unwrapped image.

We have tested our approach on various software-implemented QUBO solvers and the D-wave 2000Q annealer, and for a variety of synthetic and real images. The solutions derived by the single-pass super-pixel are either identical to the ground truth or have less than 5% of pixels differing from the ground truth. The results of the multipass super-pixel showed it can handle bigger images, while preserving the quality of the solution. The accuracy of the results depends on the specific annealer employed, the amount of noise, and the complexity of the images. Both the one-pass or two-pass super-pixel algorithms produce excellent



**FIGURE 15. Sample synthetic image with high-complexity and high-noise summarizing the effectiveness index of different approaches. From the left to right, the bars show, the ground truth, the direct TRWS applied with no super-pixel, the super-pixel one-pass solution using TRWS as subimage solver with a margin of 2 and 0, respectively, the super-pixel two-pass solution using TRWS as subimage solver with a margin of 2 and 0, respectively, and the super-pixel two-pass solution with a margin of 0 using D-wave as subimage solver.**



**FIGURE 16. Sample synthetic image with high-complexity and high-noise summarizing the effectiveness index of different approaches. From the left to right, the bars show, the ground truth, the direct TRWS applied with no super-pixel, the super-pixel one-pass solution using TRWS as subimage solver with a margin of 2 and 0, respectively, the super-pixel two-pass solution using TRWS as subimage solver with a margin of 2 and 0, respectively, and the super-pixel two-pass solution with a margin of 0 using D-wave as subimage solver.**

results for low- and medium-complexity images. Real SAR and InSAR images are typically of low or medium complexity. Our approach produces acceptable results for high-complexity images. Moreover, our experiments have shown that introducing margins indeed improves the quality of the resulting unwrapped images for both the single and the multipass super-pixel methods. Software-implemented QUBO solvers showed slightly better results compared to the D-wave 2000Q annealer. However, both achieved state-of-the-art solution quality.

Figs. 15 and 16 are examples showcasing and summarizing the results we have presented. Fig. 15 shows the results for a synthetic image selected at random with high complexity and high noise, while Fig. 16 shows the results of a synthetic image with low-complexity and low-noise.

Fig. 15 shows that the introduction of margins improves the performance (third and fourth columns, and fifth and sixth columns). It also shows that the two-pass has slightly

inferior results compared to the one-pass because it is more susceptible to aliasing.

The results in Fig. 16 are mostly of 100% effectiveness index. This confirms that the complexity and noise affect the results and we can speculate that in the case of high noise/high complexity the Nyquist condition is violated [11].

For both cases, the D-wave annealer (D-wave 2000Q) we used showed slightly inferior but comparable performance to the TRWS-based methods.

We expect subsequent developments and improvements by D-wave to yield better results. While an exact analysis of the expected scaling is beyond the scope of this article, we expect significant improvements by chips with higher connectivity. Next generation solver technology is expected to improve or even eliminate the need to manually encode integer variables into binary ones.

## APPENDIX A COST DERIVATION

Denoting by  $\varphi_i$  the phase of pixel  $i$ , and by  $\phi_i$  the wrapped phase of the same pixel, we can relate the phase and wrapped phases of pixels  $i$  and  $j$  as follows:

$$\varphi_i = \phi_i + 2\pi k_i \quad (21)$$

and

$$\varphi_j = \phi_j + 2\pi k_j. \quad (22)$$

Further, due to the Nyquist criterion, and if pixels  $i$  and  $j$  are neighboring, then

$$\varphi_i - \varphi_j < 2\pi. \quad (23)$$

The wrap function is defined as

$$\text{wrap}(\theta) = \arg(e^{i\theta}) = \theta - \left\lfloor \frac{\theta}{2\pi} \right\rfloor. \quad (24)$$

Then, we can reason as follows: From (21) and (22), we have

$$\varphi_i - \varphi_j = \phi_i - \phi_j + 2\pi(k_i - k_j) \quad (25)$$

or, applying the  $\text{wrap}(\cdot)$  function on both sides, we obtain

$$\begin{aligned} \text{wrap}(\varphi_i - \varphi_j) &= \text{wrap}(\phi_i - \phi_j + 2\pi(k_i - k_j)) \\ &\Rightarrow \varphi_i - \varphi_j - 2\pi \left\lfloor \frac{\varphi_i - \varphi_j}{2\pi} \right\rfloor \\ &= \phi_i - \phi_j + 2\pi(k_i - k_j) \\ &\quad - 2\pi \left\lfloor \frac{\phi_i - \phi_j + 2\pi(k_i - k_j)}{2\pi} \right\rfloor. \end{aligned} \quad (26)$$

Because of the Nyquist assumption [c.f. (23)]

$$\left\lfloor \frac{\varphi_i - \varphi_j}{2\pi} \right\rfloor = 0$$

and, therefore, (26) can be written as

$$\varphi_i - \varphi_j = \phi_i - \phi_j + 2\pi(k_i - k_j)$$

$$\begin{aligned} &- 2\pi \left\lfloor \frac{\phi_i - \phi_j + 2\pi(k_i - k_j)}{2\pi} \right\rfloor \\ \Rightarrow \varphi_i - \varphi_j &= \phi_i - \phi_j + 2\pi(k_i - k_j) \\ &- 2\pi \left\lfloor \frac{\phi_i - \phi_j}{2\pi} + (k_i - k_j) \right\rfloor \Rightarrow \\ \varphi_i - \varphi_j &= \phi_i - \phi_j + 2\pi(k_i - k_j) \\ &- 2\pi \left\lfloor \frac{\phi_i - \phi_j}{2\pi} \right\rfloor - 2\pi(k_i - k_j) \end{aligned}$$

as  $(k_i - k_j)$  is an integer. Therefore

$$\varphi_i - \varphi_j = \phi_i - \phi_j - 2\pi \left\lfloor \frac{\phi_i - \phi_j}{2\pi} \right\rfloor = \text{wrap}(\phi_i - \phi_j). \quad (27)$$

Using (27) and (25), we obtain

$$\begin{aligned} \varphi_i - \varphi_j &= \phi_i - \phi_j + 2\pi[k_i - k_j] = \text{wrap}(\phi_i - \phi_j) \\ \Rightarrow k_i - k_j &= \frac{\text{wrap}(\phi_i - \phi_j) - (\phi_i - \phi_j)}{2\pi}. \end{aligned} \quad (28)$$

Denoting

$$a_{ij} \stackrel{\text{def}}{=} \frac{\text{wrap}(\phi_i - \phi_j) - (\phi_i - \phi_j)}{2\pi}. \quad (29)$$

Equation (28) is written as

$$k_i - k_j = a_{ij} \Rightarrow k_i - k_j - a_{ij} = 0. \quad (30)$$

This equation is the basis of the cost function the optimization of which will produce appropriate values for the labels  $k_i$ .

The unwrapping problem can then be expressed as an optimization problem of the cost function

$$E = \sum_{(s,t) \in A} W_{st} |k_t - k_s - a_{st}| \quad (31)$$

that is

$$\arg \min_k \left[ \sum_{(s,t) \in A} W_{st} |k_t - k_s - a_{st}| \right] \quad (32)$$

where  $k_i$  are the labels that will determine the original phase as per (1),  $A$  is the set of pixels in the SAR image, and  $W_{st}$  are weights defining the neighborhood structure.

## APPENDIX B MQIO RESULTS OF EVALUATING THE SUPER-PIXEL ALGORITHM

Herein, we extend the results presented in Section V-B to include the simulated annealing and the parallel tempering solvers. The results are shown in Figs. 17 and 18.

### A. DISCUSSION

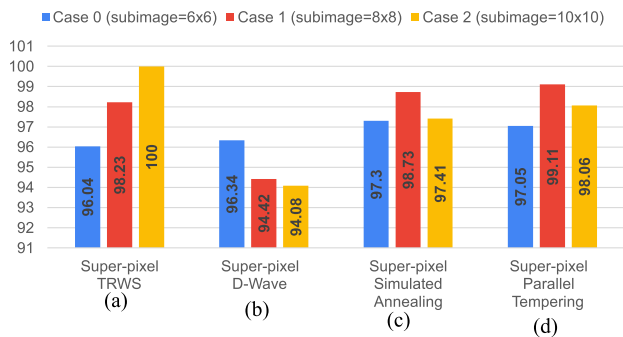
Using TRWS as a subimage solver for the super-pixel algorithm gives results that are close to the ground truth. The QUBO solvers, the MQIO solvers and the D-wave annealer,

**TABLE 12. Results of Adding Margin While Fixing the Total Subimage Size With QUBO Solvers and a Synthetic Image**

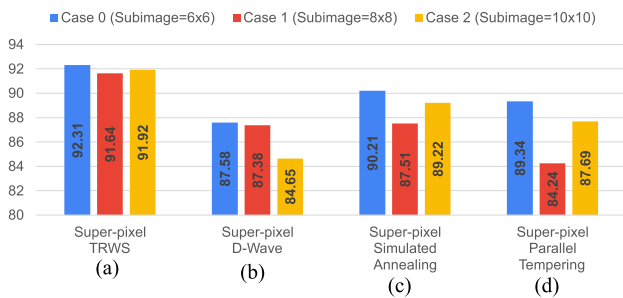
Experiment	Sub-image	margin	TRWS (Mean/STD)	Super-pixel		
				D-Wave	Simulated Annealing	Parallel Tempering
Case 0	6 × 6	2	97.83/1.01	92.16	97.51	97.87
Case 1	8 × 8	1	98.15/1.00	92.07	98.21	98.25
Case 2	10 × 10	0	97.16/1.03	93.57	96.89	97.53

**TABLE 13. Results of Adding Margin While Fixing the Total Subimage Size With QUBO Solvers and a Real Image**

Experiment	Sub-image	margin	TRWS (Mean/STD)	Super-pixel		
				D-Wave	Simulated Annealing	Parallel Tempering
Case 0	6 × 6	2	95.38/1.28	90.15	88.36	89.58
Case 1	8 × 8	1	93.85/1.30	90.84	85.8	86.82
Case 2	10 × 10	0	91.92/1.27	84.65	89.22	87.69



**FIGURE 17. Super-pixel effectiveness index for a synthetic image for the different subimage sizes and different solvers (TRWS, D-wave, simulated annealing, and parallel tempering).**



**FIGURE 18. Super-pixel effectiveness index for a real image for the different subimage sizes and different solvers (TRWS, D-wave, simulated annealing, and parallel tempering).**

provide close results to the TRWS results when used as subimages solvers. Particularly close are the results obtained by the MQIO solvers (simulated annealing and parallel tempering), while the ones from the D-wave annealer have a slightly lower effectiveness index (about 3 percentage points) as compared to the effectiveness indexes obtained by the TRWS, and the MQIO solvers.

One cannot derive any strong rules from the results shown since our single image experiments are not statistically robust. However, there are some trends, which could be the basis for future research. The effectivenesses of the D-wave solutions are slightly inferior to those of the MQIO solvers.

This may be a manifestation of decoherence. Additionally, larger subimages tend to produce superior results. However, when QUBO solvers are used, larger subimages present a much larger search space and hence a QUBO solver may not be able to reach an optimum within a reasonable time. This is manifested in Fig. 17, where the use of medium-sized subimages (8 × 8) yields better results as compared to the other two cases.

### APPENDIX C MQIO RESULTS EVALUATION OF THE ADDITION OF MARGINS TO THE SUBIMAGES

Herein, we extend the results presented in Section VI-B to include the simulated annealing and the parallel tempering solvers. The results are shown in Tables 12 and 13.

#### A. DISCUSSION

Similar to the discussion in Appendix B, one cannot derive any strong general rules of the behavior of the different solvers since our single-image experiments are not statistically robust. Notable is the trend that the introduction of a margin seems to benefit the efficacy of the unwrapping in most of the cases, the exception being that of D-wave for the single synthetic image (Table 12). Different images and different algorithms seem to be affected by the different margin sizes, e.g., the simulated annealing was the best for no margin (case 2) for the real image (Table 13). While parallel tempering has its best performance for a margin of 1 for the synthetic image (Table 12, case 1). Although larger margins seem to benefit the efficacy of the solution, larger images, due to the larger search space, seem to work against the gain realized by the introduction of margins.

#### ACKNOWLEDGMENT

The authors would like to thank 3vGeomatics for the InSAR images and software used in this research

#### REFERENCES

[1] J. Cai, W. G. Macready, and A. Roy, "A practical heuristic for finding graph minors," 2014, doi: [10.48550/arXiv.1406.2741](https://doi.org/10.48550/arXiv.1406.2741).

- [2] C. W. Chen and H. A. Zebker, "Phase unwrapping for large SAR interferograms: Statistical segmentation and generalized network models," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 8, pp. 1709–1719, Aug. 2002, doi: [10.1109/TGRS.2002.802453](https://doi.org/10.1109/TGRS.2002.802453).
- [3] M. Costantini, "A novel phase unwrapping method based on network programming," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 3, pp. 813–821, May 1998, doi: [10.1109/36.673674](https://doi.org/10.1109/36.673674).
- [4] C. J. Curlander and R. N. McDonough. *Synthetic Aperture Radar-Systems and Signal Processing*. New York, NY, USA: Wiley, 1991.
- [5] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, "Quantum annealing: A new method for minimizing multidimensional functions," *Chem. Phys. Lett.*, vol. 219, no. 5-6, pp. 343–348, 1994, doi: [10.1016/0009-2614\(94\)00117-0](https://doi.org/10.1016/0009-2614(94)00117-0).
- [6] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984, doi: [10.1109/TPAMI.1984.4767596](https://doi.org/10.1109/TPAMI.1984.4767596).
- [7] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," 2018, doi: [10.48550/arXiv.1811.11538](https://doi.org/10.48550/arXiv.1811.11538).
- [8] B. Heim, T. F. Rønnow, S. V. Isakov, and M. Troyer, "Quantum versus classical annealing of Ising spin glasses," *Science*, vol. 348, no. 6231, pp. 215–217, 2015, doi: [10.1126/science.aaa4170](https://doi.org/10.1126/science.aaa4170).
- [9] M. W. Johnson *et al.*, "Quantum annealing with manufactured spin," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011, doi: [10.1038/nature10012](https://doi.org/10.1038/nature10012).
- [10] A. Jooya, B. Keshavarz, N. Dimopoulos, and J. S. Oberoi, "Accelerating neural network ensemble learning using optimization and quantum annealing techniques," in *Proc. 2nd Int. Workshop Post Moores Era Supercomputing*, 2017, pp. 1–7, doi: [10.1145/3149526.3149528](https://doi.org/10.1145/3149526.3149528).
- [11] K. A. H. Kelany, "Deep learning and quantum annealing methods in synthetic aperture radar," Ph.D. thesis, Univ. Victoria, Victoria, B.C., Canada, 2021.
- [12] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006, doi: [10.1109/TPAMI.2006.200](https://doi.org/10.1109/TPAMI.2006.200).
- [13] A. M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [14] R. H. Swendsen and J.-S. Wang, "Replica Monte Carlo simulation of spin-glasses," *Phys. Rev. Lett.*, vol. 57, no. 21, 1986, Art. no. 2607, doi: [10.1103/PhysRevLett.57.2607](https://doi.org/10.1103/PhysRevLett.57.2607).
- [15] J. S. Yedidia, "Message-passing algorithms for inference and optimization," *J. Stat. Phys.*, vol. 145, no. 4, pp. 860–890, 2011, doi: [10.1007/s10955-011-0384-7](https://doi.org/10.1007/s10955-011-0384-7).
- [16] A. Zaribafian, D. J. J. Marchand, and S. S. C. Rezaei, "Systematic and deterministic graph minor embedding for Cartesian products of graphs," *Quantum Inf. Process.*, vol. 16, no. 5, pp. 1–26, 2017, doi: [10.1007/s11128-017-1569-z](https://doi.org/10.1007/s11128-017-1569-z).



**Khaled A. Helal Kelany** was born in Cairo, Egypt, in 1989. He received the B.Eng. in electronics and electrical communications and M.Sc. degree from Cairo University, Giza, Egypt, in 2013 and 2016, respectively, and the Ph.D degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, in 2021.

He joined the Department of Electrical and Computer Engineering, University of Victoria, in 2017. His research interests include quantum

computing, artificial neural networks, parallel computing, and computer architecture.

Dr. Kelany was the recipient of the IEEE International Conference on Quantum Computing and Engineering Best Paper Award, in 2020.



**Nikitas Dimopoulos** (Life Senior Member, IEEE) received the B.Sc. degree in physics from the University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 1975, 1976 and 1980 respectively.

He joined the Department of Electrical and Computer Engineering, University of Victoria (UVic.), Victoria, BC, Canada, in 1988, where he is currently Professor and Lansdowne Chair of Computer Engineering. He served as the Chair of

the Department, from 1998 to 2003 and 2005 and 2008, and was Visiting Professor with the Computer Engineering Laboratory, Delft University of Technology, Delft, The Netherlands, in 2001. Previous to his appointment at UVic, he was an Assistant and then an Associate Professor with the Department of Electrical Engineering, Concordia University, Montreal, QC, Canada, from 1980 to 1987, and member of the Technical Staff with the Jet Propulsion Laboratory, Pasadena, CA, USA, from 1986 to 1987. His research has been funded by the Natural Sciences and Engineering Research Council, Canadian Cable Labs Fund, The Canada Foundation for Innovation, and the Canadian Microelectronics Corporation (CMC Microsystems). He has authored or coauthored 200 publications including edited volumes, refereed journal, conference papers, and technical reports. His research interests include computer architecture, neural networks, and quantum computing.

Prof. Dimopoulos is a Professional Engineer registered with Engineers and Geoscientists in British Columbia. He is a Fellow of the Engineering Institute of Canada.



**Clemens P. J. Adolphs** received the graduate degree in computer science and physics from the RWTH Aachen University of Technology, Aachen, Germany, the Ph.D. degree in physics, with a focus on theoretical condensed matter physics, from the University of British Columbia, Vancouver, BC, Canada, in 2016.

In 2016, he joined IQbit, Vancouver, BC, Canada, where he is Senior Optimization Solutions Lead. While at IQbit, he has been involved in multiple projects exploring quantum-inspired

solutions for business cases with clients in finance, insurance, and logistics. His research interests include the development of quantum and classical solutions and applying agile software development techniques.



**Amirali Baniasadi** received the B.Sc. degree in electronic engineering from Tehran University, Tehran, Iran, and the M.Sc. degree in digital electronics engineering from Sharif University of Technology, Tehran, Iran, in 1992 and 1995, respectively, and the Ph.D. degree in computer engineering from Northwestern University, Illinois, Evanston, IL, USA, in 2002.

In 2002, he joined the Department of Electrical and Computer Engineering, University of Victoria (U.Vic.), Victoria, BC, Canada, where he is

currently a Professor. He was Visiting Professor with the Sharif University of Technology, Tehran, Iran, in 2007, and Stanford University, Stanford, CA, USA, from 2016 and 2017. His research has been funded by the Natural Sciences and Engineering Research Council of Canada, Consortium for Aerospace Research and Innovation in Canada, and MITACS (a Canadian Network of Centres of Excellence). He has authored or coauthored 100 publications including refereed journal, conference papers, and technical reports. His research interests include computer architecture, neural networks, and quantum computing.