

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/277557601>

# Multichannel Phase Unwrapping: Problem Topology and Dual-Level Parallel Computational Model

Article in IEEE Transactions on Geoscience and Remote Sensing · October 2015

DOI: 10.1109/TGRS.2015.2430619

CITATIONS

15

READS

856

3 authors, including:



**Pasquale Imperatore**

Italian National Research Council

88 PUBLICATIONS 801 CITATIONS

[SEE PROFILE](#)



**A. Pepe**

Italian National Research Council

204 PUBLICATIONS 4,289 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Special Issue "SAR Interferometry: Methods and Applications for Earth Science and Environmental Monitoring" [View project](#)



Special Issue "Monitoring Sustainable Development Goals" ([www.mdpi.com/si/37312](http://www.mdpi.com/si/37312)) [View project](#)

# Multichannel Phase Unwrapping: Problem Topology and Dual-Level Parallel Computational Model

Pasquale Imperatore, *Member, IEEE*, Antonio Pepe, *Member, IEEE*, and Riccardo Lanari, *Fellow, IEEE*

**Abstract**—In the theoretical purview of the discrete Calculus, a rigorous gradient-based formulation of the multichannel phase unwrapping (MCh-PhU) problem is systematically established in terms of discrete differential operators, which are defined by the topology of the intrinsically discrete spaces upon which they act, thus capturing the essential topological character of the problem within a suitable matrix formalism and providing interesting implications. Within this methodological framework, the extended minimum cost flow (EMCF) algorithm, which provides an effective strategy aimed at solving the MCh-PhU problem, is revised, and its computational structure is analyzed. A parallel formulation of the computational-intensive EMCF algorithm is then presented. Emphasis is placed on the methodological and practical aspects leading to a novel dual-level parallel computational model in which the parallelism is hierarchically implemented at two different levels. Performance evaluation relevant to the implemented prototype solution is also carried out, thus quantifying the benefit of parallelism at different levels. The significant experimentally achieved speedup demonstrates the validity of our approach. As a result, the attained parallel prototype enables the large-scale solution of the MCh-PhU problem in a reasonable time frame, with a great impact on systematic exploitation of the available SAR archives.

**Index Terms**—Discrete calculus, high-performance computing (HPC), parallel computing, phase unwrapping (PhU), synthetic aperture radar interferometry (InSAR).

## I. INTRODUCTION AND MOTIVATIONS

**P**HASE unwrapping has been a subject of great interest in different disciplines in which coherent processing is concerned (magnetic resonance imaging, synthetic aperture radar/sonar, optical interferometry, diffraction tomography, astronomical imaging, holographic interferometry, etc.) [1], and, in particular, in synthetic aperture radar (SAR) interferometry (InSAR) [2]–[5]. In all of these cases, the phase can be related to some physical quantity of interest.

Phase unwrapping (PhU) essentially consists in recovering the absolute (unwrapped) phase field from the observed (wrapped) modulo- $2\pi$  phase value. Nevertheless, the abso-

lute phase estimation is a nonlinear inverse problem, which is mathematically ill-posed in the sense of *Hadamard* (the uniqueness requirement is violated) if no further information or assumptions are added. Moreover, within the InSAR context, the PhU problem solution is exacerbated by noise, discontinuities, and phase aliasing effects, which are ascribable to radar shadow, foreshortening, layover, and decorrelation phenomena, thus causing inconsistencies in the phase (i.e., the measured “pseudo-gradient” field is nonconservative) [1], [6]–[16].

The solution of the *Multichannel PhU* (MCh-PhU) problem is generally required in multichannel InSAR techniques. In particular, multichannel (or multitemporal) InSAR techniques address the problem of combined processing of multiple SAR acquisitions over the same area. These techniques can be essentially categorized in two main classes: *permanent scatterer* (PS)-based techniques, which rely on temporally coherent point-wise scatterers [17]–[20] only, and *small baseline* (SB)-based techniques, which resort to interferometric data pairs with baselines below a critical value [21]–[24].

Within this context, MCh-PhU procedures, which are primarily aimed at estimating and characterizing time-dependent Earth’s surface deformation processes, have successfully been exploited and largely discussed in the literature [3], [25]–[28]. However, they exhibit some important limitations that deserve to be discussed.

First, the MCh-PhU procedure constitutes the most critical task of the multichannel InSAR techniques from a computational point of view, due to its extremely time-consuming nature [10], [21], [25]. This point is then crucial, whenever the increasingly available large-volume multidimensional SAR data, conveying information about complex Earth’s crust events, have to be systematically investigated on suitable space–time scales for geospatial phenomena understanding. Accordingly, MCh-PhU practical application to large data sets poses some challenges, due to the associated intensive computation and storage requirements.

Secondly, we highlight that the gradient-based formulations of MCh-PhU approaches typically rely on a discrete-setting-based description [1], [7], [29]; however, they manifestly resort to the concepts of gradient and curl of the conventional vectorial calculus, which inherently imply a reference to an underlying continuum space and the notion of the infinitesimal. This evident inconsistency poses some conceptual limitations from a mathematical viewpoint, and accordingly, the existing MCh-PhU formulations generally lack of a rigorous mathematical framework.

To overcome the implicit limitations inherent to the particularly time-consuming nature of the sequential MCh-PhU

Manuscript received December 12, 2014; revised February 26, 2015; accepted May 2, 2015. This work was supported in part by the Italian Space Agency, by the Italian Department of Civil Protection, by the Italian Ministero dell’Istruzione, dell’Università e della Ricerca (MIUR) under Project “Studio multidisciplinare della fase preparatoria di un terremoto,” and by the European Commission under the MED-SUV Project (European Union’s Seventh Program for research, technological development and demonstration) under Grant 308665.

The authors are with the Istituto per il Rilevamento Elettromagnetico dell’Ambiente (IREA), National Research Council (CNR), 80124 Napoli, Italy (e-mail: imperatore.p@irea.cnr.it; pepe.a@irea.cnr.it; lanari.r@irea.cnr.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2015.2430619

algorithms and fully address relevant large-scale computations in a reasonable time frame, without sacrificing the accuracy, the exploitation of the modern parallel computational platforms seems particularly appropriate. Accordingly, computational demanding MCh-PhU algorithms motivate the devising of specific parallel computational models and the adoption of appropriate methodologies and tools, to exploit the available parallelism. However, developing a relevant parallel model for a specific (sequential) computation-intensive scientific application, in a form that allows benefiting from the multiple processing units, remains a challenging problem. Likewise, the development of *scalable* algorithms for implementation on parallel processing architectures has become an important research topic in scientific computing over the past few years [30]–[41].

On the other hand, overcoming the difficulty inherent in the mentioned mathematical inconsistencies demands new formal approaches. From a methodological perspective, a rigorous mathematical framework is highly desirable since it would provide an unambiguous and theoretical-consistent formalism for the MCh-PhU problem formulation.

In this paper, we focus on two different main issues. Primarily, the fundamental tools of *discrete calculus* are introduced and properly phrased on a discrete space [42]. Emphasis is placed on the topological characterization of the underlying discrete setting provided by the *differential operators* of the discrete calculus, which are formally associated with matrix operators and represent the discrete counterparts of the classical differential operators of the vector calculus. A formulation of the MCh-PhU problem, consistent with the theoretical foundation of the discrete calculus, is then provided via a systematic matrix formalism. In addition, unlike previous approaches, such a formal framework enables meaningful analytical investigations on a mathematical consistent playground, also permitting to express previous obtained results in a more general way. Within this framework, the *extended minimum cost flow* (EMCF) algorithm [25], which provides an effective strategy aimed at solving the MCh-PhU problem, is revised and its inherent computational structure is detailed. The rationale of a parallel reformulation for the EMCF algorithm is then presented, emphasizing the methodological and practical aspects leading to a novel *dual-level* parallel computational model, in which the parallelism is hierarchically implemented at two different levels. The parallelization strategy is clarified, and the adopted criteria are fully discussed in this paper.

This paper is organized as follows. In Section II, the discrete-calculus basic concepts are presented, and accordingly, the MCh-PhU problem is suitably formalized. EMCF algorithm is formally reviewed in Section III, also emphasizing its inherent computational aspects. The novel parallel computational model is described in Section IV. The design of the relevant prototype implementation is outlined in Section V. Experimental analysis and pertinent performance evaluation, aimed at quantifying the benefit of parallelism, are carried out and discussed in Section VI.

Some remarks are now in order to preliminarily establish the basic notation used throughout this paper. The “hat” operator is used to indicate an estimate, e.g.,  $\hat{\mathbf{z}}$  is an estimate of  $\mathbf{z}$ .

The transpose operator is denoted with superscript  $T$ . Matrices, vectors, and scalars are represented by italic bold uppercase symbols, bold lowercase symbols, and lowercase symbols, respectively. A  $Q \times P$  matrix  $\mathbf{A}$  is then represented as

$$\mathbf{A} = [a_{qp}] = [\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^P] = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_Q \end{bmatrix}$$

where its  $p$ th column is denoted by the  $Q$ -dimensional vector  $\mathbf{a}^p$  and its  $q$ th row by the  $P$ -dimensional vector  $\mathbf{a}_q$ , and the matrix entries are indicated with  $a_{qp}$ , with  $q \in \{1, 2, \dots, Q\}$  and  $p \in \{1, 2, \dots, P\}$ .

## II. MCh-PHU PROBLEM WITHIN THE PURVIEW OF THE DISCRETE CALCULUS

In this section, we review the mathematical formulation of the MCh-PhU problem within the purview of the discrete calculus. As a matter of fact, a graph-based description naturally arises in formulating the MCh-PhU problem, due to the underlying discrete irregular data structure. Indeed, as far as discrete settings (e.g., graphs) are concerned, resorting to the conventional vectorial calculus might not be adequate since it inherently implies a reference to an underlying continuum space. On the contrary, *discrete calculus* offers a rigorous methodological framework since it treats a discrete domain entirely as its own entity. In particular, discrete calculus provides proper differential operators that make it possible to purely operate onto a finite discrete structure without referring to the continuous space and notion of the infinitesimal [42]. More specifically, the introduction of some well-known algebraic structures [42]–[45], capturing the essential topological character of the underlying graphs, permits to phrase pertinent *differential operators* as matrices. Therefore, one of the most important consequences of this approach is that the purely topological nature of the discrete differential operators is made more apparent and concrete. Accordingly, by systematically adopting the relevant key concepts and tools available within this theoretical framework, we here provide a description of the MCh-PhU problem on a suitable mathematical playground. For such a purpose, we first establish the notation and terminology used throughout the subsequent sections (Section II-A), and then the problem at hand is reviewed within this formalism (Section II-B and C). We remark that the focus here is on presenting key concepts that are useful for the following analyses; however, a comprehensive treatment of the discrete calculus and related huge fields of mathematics (e.g., algebraic topology, exterior calculus, and differential forms) is clearly beyond the scope of this paper but can be found in [42]–[45].

### A. Differential Operators of the Discrete Calculus

A graph  $\mathcal{G} = (\mathcal{V}; \mathcal{E})$  is defined by two sets:  $\mathcal{V}$  and  $\mathcal{E}$ . The former is the set of nodes (or vertices) of the graph, and the latter represents the corresponding set of edges. Let  $Q$  and  $M$  be the cardinality of  $\mathcal{V}$  and  $\mathcal{E}$ , respectively. The vector space

$\mathbb{R}^M$  is referred to as the *edge space* and the vector space  $\mathbb{R}^Q$  is referred to as the *vertex space*, with  $\mathbb{R}$  denoting the field of real numbers. Without loss of generality, we here assume that the graph  $\mathcal{G}$  is *connected* (i.e., every pair of vertices in the graph is connected [45]). Moreover, an orientation establishes a default direction on an edge that is considered positive or negative, thus yielding an oriented graph.

The  $M \times Q$  incidence matrix  $\mathbf{II} = [\mathbf{II}_{mq}]$  of an oriented graph  $\mathcal{G}$  specifies its edge-node connectivity relations, whose entries are defined as [43], [45]

$$\mathbf{II}_{mq} \equiv \begin{cases} -1, & \text{if } m\text{th edge starts at } q\text{th node} \\ +1, & \text{if } m\text{th edge ends at } q\text{th node} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

with  $m = 1, 2, \dots, M$  and  $q = 1, 2, \dots, Q$ . It is important to note that the column rank of  $\mathbf{II}$  is  $Q - 1$ . The incidence matrix  $\mathbf{II}$  generates an orthogonal decomposition  $\mathcal{R}(\mathbf{II}) \oplus \mathcal{N}(\mathbf{II})^T = \mathbb{R}^M$ , where  $\mathcal{R}(\mathbf{II})$  is the column space of the incidence matrix  $\mathbf{II}$ , and  $\mathcal{N}(\mathbf{II}^T)$  denotes the kernel (or null-space) of the matrix  $\mathbf{II}^T$ .

The notion of cycle space is also fundamental in graph theory. The *cycle space* of the graph  $\mathcal{G}$ , namely  $\mathcal{C} = \mathcal{C}(\mathcal{G})$ , is the subspace of edge space  $\mathbb{R}^M$  spanned by all the cycles in  $\mathcal{G}$ . The dimension of  $\mathcal{C}(\mathcal{G})$  is also referred to as the *cyclomatic number of  $\mathcal{G}$*  [45]. It is also well known that, for every *connected* graph  $\mathcal{G}$  with  $Q$  nodes and  $M$  edges, the dimension of the cycle space is given by  $R = \dim(\mathcal{C}(\mathcal{G})) = M - Q + 1$  [42]. Each basis for  $\mathcal{C}(\mathcal{G})$  (i.e., the cycle basis) is, therefore, uniquely specified by a  $M \times R$  matrix  $\mathbf{\Omega}$ , which is called *cycle matrix*. Thus, the column vectors of  $\mathbf{\Omega} = [\boldsymbol{\omega}^1, \dots, \boldsymbol{\omega}^R]$  form a basis for a  $R$ -dimensional vector subspace (the cycle space of  $\mathcal{C}(\mathcal{G})$ ) of  $\mathbb{R}^M$ .  $\mathcal{C}(\mathcal{G})$  is indeed the null space of  $\mathbf{II}^T$ , so that a cycle basis provides a basis for  $\mathcal{N}(\mathbf{II}^T)$  [42]. Accordingly, a fundamental property of a linear graph is expressed by the following remarkable relations:

$$\mathbf{II}^T \mathbf{\Omega} = \mathbf{0} \quad (2)$$

$$\mathbf{\Omega}^T \mathbf{II} = \mathbf{0}. \quad (3)$$

Indeed, several methods for defining a cycle set have been studied, and they can be used to define incidence relations between edges and cycles. Specifically, the definition of a cycle set from the edge set can be obtained algebraically and geometrically (i.e., from an embedding). Algebraic methods find a suitable  $M \times R$  matrix  $\mathbf{\Omega}$  whose columns provide a basis for the null space of  $\mathbf{II}^T$ , with  $R = \dim(\mathcal{N}(\mathbf{II}^T))$ . Geometric methods for defining a cycle set (i.e., from an embedding) permit to identify algorithmically a cycle set (representing the faces) in this embedding and may be used to produce the edge-face incidence matrix  $\mathbf{\Omega}$  (as illustrated in Fig. 1). In particular, it is possible to consider a normalized irreducible cycle basis forming elementary (or irreducible) cycles [42]–[45], i.e., cycles that contain no other cycles, so that we can associate

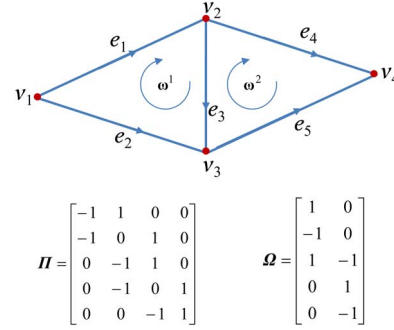


Fig. 1. Example graph shown along with its edge-node incidence matrix  $\mathbf{II}$  and cycle (edge-face incidence) matrix  $\mathbf{\Omega}$ . Note that  $M = 5$ ,  $Q = 4$ , and  $R = 2$ .

to each elementary cycle an elementary cycle vector  $\boldsymbol{\omega}^r = [\omega_1, \dots, \omega_M]^T$ , whose entries are defined as

$$\omega_i \equiv \begin{cases} +1, & \text{if } r\text{th cycle traverses } i\text{th edge forward} \\ -1, & \text{if } r\text{th cycle traverses } i\text{th edge backward} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Accordingly, the so defined  $\mathbf{\Omega} = [\boldsymbol{\omega}^1, \dots, \boldsymbol{\omega}^R]$  provides a particularly attractive basis for  $\mathcal{N}(\mathbf{II}^T)$ , i.e., the cycle basis formed by all the elementary cycle vectors associated with the elementary cycles in  $\mathcal{G}$ . Accordingly, we note that  $\mathbf{\Omega}$  defines the incidence connectivity relations between edges and cycles (see Fig. 1).

It is instructive to highlight that the topological operators  $\mathbf{II}$ ,  $\mathbf{II}^T$ , and  $\mathbf{\Omega}$  provide the discrete counterparts of the classical *gradient* ( $\nabla$ ), *divergence* ( $\nabla \cdot$ ), and *curl* ( $\nabla \times$ ) operators of the vector calculus for continuous space, respectively. Accordingly, they can be regarded as differential operators on the discrete setting [42]. In addition, it is worth emphasizing that the identities (2) and (3) mimic the properties of their classical vector calculus counterparts  $\nabla \cdot \nabla \times = 0$  ( $\text{div curl} = 0$ ) and  $\nabla \times \nabla = 0$  ( $\text{curl grad} = 0$ ), respectively. It should be pointed out that  $\mathbf{II}$  yields differences along edges of nodal “potentials” represented by  $\mathbf{II} \mathbf{x}$ . Conversely, given an arbitrary  $\mathbf{f} \in \mathbb{R}^M$ , a solution of the equation  $\mathbf{II} \mathbf{x} = \mathbf{f}$  (if it exists) is called the *potential of  $\mathbf{f}$* . Note also that  $\mathbf{x}$  (if it exists) is not unique since the constant column  $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^Q$  is an element of the kernel of  $\mathbf{II}$ . Of course, not every  $\mathbf{f} \in \mathbb{R}^M$  is the discrete gradient of some  $\mathbf{x}$  since  $\mathbf{f}$  may contain a curl component. Indeed, a prescribed  $\mathbf{f} \in \mathbb{R}^M$  can be written as a nodal difference ( $\mathbf{f} = \mathbf{II} \mathbf{x}$ ) if it is *cyclically consistent*, i.e., if it satisfies  $\mathbf{\Omega}^T \mathbf{f} = \mathbf{0}$  (i.e., there is no component of the flow in the cycle space). Note also that  $\mathbf{\Omega}$  is the (cross) differential operator of the graph whose expression can be given in terms of a normalized cycle basis;  $\mathcal{N}(\mathbf{\Omega}^T)$  denotes the subspace of  $\mathbb{R}^M$  with zero flow circulation (curl free) around cycles. Moreover,  $\mathbf{II}^T \mathbf{f}$  yields nodal accumulations from flows along edges.

As a result, the differential operators, as basic tools of the discrete calculus, have been established and properly phrased on the discrete space. This mathematical abstraction meaningfully captures the topological structure of the underlying discrete setting. Note that the topological characterization of the graph is

essentially embodied in the algebraic structure of the associated discrete (matrix) operators and their interrelations. We also stress the significant distinction between the discrete-gradient operator and the commonly adopted discretized version of the continuous gradient obtained via the method of finite differences in numerical analysis; the latter does lack the desirable topological behavior [42].

As a final remark, some considerations on the *total unimodularity* (TU) property inherent to the matrix operators, which is extremely important in integer linear programming, are in order. We recall that a matrix is TU if the determinant of every submatrix is either 0 or  $\pm 1$ . For any graph, the edge-node incidence matrix is TU. On the contrary, the face-edge incidence matrix, in general, is not TU [42], [46]. However, the edge-face incidence matrix is TU when each edge is included in exactly two faces that traverse the edge in opposite directions (e.g., a planar graph with a minimum cycle basis [42]). In this circumstance, TU of the edge-face incidence matrix stems from the fact that the face-edge incidence matrix is the edge-node incidence matrix of the dual graph (see also Section III-D and [42]). Indeed, a TU constraint matrix (and integer constraints) guarantees that the solution of the related optimization problem (see also the optimization problem (21) and (22) in the following) will be integer. Nonetheless, TU property has a further practical significance since the relaxed problem, which is obtained by neglecting the integer constraints, can also be solved using generic linear (not integer) programming solvers.

### B. Statement of the MCh-PhU Problem

Once the basic concepts of the discrete calculus and graph theory have been presented, we are now in position to frame the formulation of the MCh-PhU problem on an appropriate mathematical playground. Let us consider a set of  $Q$  *single-look-complex* (SLC) SAR data acquired over a certain area of interest. One of them is assumed as the reference (master) image, with respect to which the images are properly coregistered. This set is characterized by the corresponding acquisition times  $\{t_1, \dots, t_Q\}$  and perpendicular baselines  $\{b_{\perp 1}, \dots, b_{\perp Q}\}$ . Accordingly, for each coregistered SLC pair, a multilooked *phase interferogram* (suitably depurated by the flat Earth and topography contributions, by using *a priori* information about the topography and acquisition geometry)<sup>1</sup> can be produced [47]; however, a common practice (within the multitemporal SB InSAR class) is first to identify a suitable small-baseline subset of the relevant multibaseline (temporal and spatial perpendicular baselines) interferometric pair set [21]. This is done to confine the effect of decorrelation phenomena associated with inherent angular and temporal electromagnetic backscattering variations [48]. Furthermore, a subset of common pixels of the  $M$  interferograms are then usually identified via the estimated *coherence* [29], [48], so that only  $P$  pixels characterized by relatively high coherence values are singled out. In other words, the coherence index, providing a quantitative estimation of the

decorrelation effects, permits to discriminate in favor of the “reliable” pixels.

The final aim is to reconstruct the absolute (i.e., not restricted in the principal  $[-\pi, \pi)$  interval) interferometric phase values from the wrapped (i.e., observed only in the principal  $[-\pi, \pi)$  interval) interferometric phase pertinent to  $M$  multichannel interferograms.

The problem we are interested in can be naturally framed on a discrete setting. Indeed, one possibility is to regard the discrete set of  $P$  selected (typically sparsely distributed) coherent pixels as a set of nodes  $\mathcal{V}_B$ , in the Euclidean (*azimuth, range*) plane, and the set of  $Q$  SAR acquisitions representing a set of nodes  $\mathcal{V}_A$  in the Euclidean (*temporal baseline, perpendicular baseline*) plane [25]. Accordingly, a formal description of the problem at hand can be given in terms of a couple of abstract graphs:  $\mathcal{G}_A = (\mathcal{V}_A; \mathcal{E}_A)$  and  $\mathcal{G}_B = (\mathcal{V}_B; \mathcal{E}_B)$ , where the corresponding edge sets  $\mathcal{E}_A$  and  $\mathcal{E}_B$  have to be properly defined. Accordingly, with  $Q$  and  $M$ , we denote the cardinality of  $\mathcal{V}_A$  and  $\mathcal{E}_A$ , and with  $P$  and  $N$ , the cardinality of  $\mathcal{V}_B$  and  $\mathcal{E}_B$ , respectively. Note also that defining  $\mathcal{E}_A$  pertains to the  $M$  interferometric pairs selection. Defining a meaningful edge set for a collection of nodes is now the concern since different criteria can be adopted to achieve it. Dimensionality of the ambient space in which the graph is embedded deserves some considerations. In this regard, we recall that a graph is called *planar* if it can be embedded in the plane [42]–[45]. Note also that a graph is not generally guaranteed to be planar even if the nodes are embedded in two-dimensions. Since planarity is important for the workability of the implicated optimization procedure with powerful numerical solvers (as discussed in Section III-D), a typical option to preserve graph planarity is resorting to the *Delaunay triangulation* in the Euclidean plane for establishing the edge set from nodes embedded in two-dimensions [25]. Note that such an option specifically pertains to the solution strategy (see Section III); nonetheless, our general formulation applies as well when different edge structures are adopted. Accordingly, once  $\mathcal{G}_A$  and  $\mathcal{G}_B$  have been somehow defined, the topological properties inherent to each graph are algebraically captured by the related differential operators, which are summarized in Table I.

First of all, we consider the absolute phase relevant to the multichannel SAR acquisition as a node variable pertinent to both the graphs  $\mathcal{G}_A$  and  $\mathcal{G}_B$ ; by using a matrix representation, this information can be conveniently arranged in a  $Q \times P$  matrix  $\Phi$  as follows:

$$\Phi = [\Phi^1, \dots, \Phi^P] = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_Q \end{bmatrix} \quad (5)$$

where  $\forall p \in \{1, 2, \dots, P\}$   $\Phi^p \in \mathbb{R}^Q$  encodes in a vectorial manner the  $p$ th node variable relevant to the graphs  $\mathcal{G}_A$ ; similarly,  $\forall q \in \{1, 2, \dots, Q\}$   $\Phi_q \in \mathbb{R}^P$  encodes the  $q$ th node variable relevant to  $\mathcal{G}_B$ .

Widely adopted global gradient-based PhU approaches, which have historically been developed for the single-channel case, generally consist of three processing steps [1], [29]. First, an estimation of the (wrapped) phase gradient is obtained; the

<sup>1</sup>Note that, by abuse of terminology, the expression “differential interferogram” is also used.

TABLE I  
ADOPTED NOTATION

Symbol	Quantity	Related meaning
$Q$	Nodes Number of $\mathcal{G}_A$	Number of SAR acquisitions
$M$	Edges Number of $\mathcal{G}_A$	Number of interferometric pairs
$R$	Dimension of the cycle space of $\mathcal{G}_A$	
$\Pi_A$	$M \times Q$ incidence matrix of $\mathcal{G}_A$	Discrete-gradient operator
$\Pi_A^T$		Discrete-divergence operator
$\Omega_A$	$M \times R$ cycle matrix of $\mathcal{G}_A$	Discrete-curl operator
$P$	Nodes Number of $\mathcal{G}_B$	Number of selected pixels
$N$	Edges Number of $\mathcal{G}_B$	
$L$	Dimension of the cycle space of $\mathcal{G}_B$	
$\Pi_B$	$N \times P$ incidence matrix of $\mathcal{G}_B$	Discrete-gradient operator
$\Pi_B^T$		Discrete-divergence operator
$\Omega_B$	$N \times L$ cycle matrix of $\mathcal{G}_B$	Discrete-curl operator

estimated phase gradient is then suitably corrected (in terms of  $2\pi$  multiples) and subsequently integrated to attain the unwrapped (absolute) phase.

Within the formulation of the MCh-PhU problem we concern [25], a twofold estimation of the discrete-gradient field is carried out onto the considered two graphs  $\mathcal{G}_A$  and  $\mathcal{G}_B$ , as discussed in the following.

The stack of the absolute interferometric phases relevant to the  $M$  (vectorized) interferograms can be formally represented through a  $P \times M$  matrix denoted by

$$\Psi = [\psi^1, \dots, \psi^M] = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_P \end{bmatrix} \quad (6)$$

where the  $P$ -dimensional vector  $\psi^m$  refers to the absolute phase field pertinent to the  $m$ th interferogram. Accordingly,  $\Psi$  is formally related to the absolute (unwrapped) phase matrix  $\Phi$  via the discrete-gradient operator  $\Pi_A$

$$\Psi^T = \Pi_A \Phi. \quad (7)$$

Note also that

$$\Psi^T = [\psi_1^T, \dots, \psi_P^T] = [\Pi_A \Phi^1, \dots, \Pi_A \Phi^P]. \quad (8)$$

By applying the discrete-gradient operator  $\Pi_B$  to the absolute phase of each interferometric pair, we obtain

$$\Pi_B \Psi = [\Pi_B \psi^1, \dots, \Pi_B \psi^M]. \quad (9)$$

As a result, by using (7), we get

$$\Pi_B \Psi = \Pi_B \Phi^T \Pi_A^T. \quad (10)$$

Second, we consider the (wrapped) interferometric phase that is uniquely defined only in the principal value range since it is obtained as the phase of a complex function. Hence, it is convenient to formally introduce the noninjective (modulo- $2\pi$ ) *wrapping operator*  $W : \varphi \in \mathbb{R} \rightarrow \text{mod}(\varphi + \pi, 2\pi) - \pi \in [-\pi, \pi)$ . It should be noted that the following trivial identities hold:

$$\begin{aligned} W(W(A) \pm W(B)) &= W(A \pm B) = W(W(A) \pm B) \\ &= W(A \pm W(B)) \end{aligned} \quad (11a)$$

$$W(A) = A + 2\pi Z \quad (11b)$$

where  $A$  and  $B$  represent two generic matrices, and  $Z$  is a suitable integer matrix.

Given the stack of the unknown absolute (unwrapped) interferometric phases  $\Psi$ , the corresponding stack of the wrapped phases  $\tilde{\Psi}$  can be conveniently expressed in terms of the *wrapped discrete gradient of (wrapped) observed phase* as follows:

$$\begin{aligned} \tilde{\Psi}^T &= [W(\Pi_A W(\Phi^1)), \dots, W(\Pi_A W(\Phi^P))] \\ &= W(\Pi_A W(\Phi)) = W(\Pi_A \Phi) \end{aligned} \quad (12)$$

where we have exploited (11b). Note also that the  $p$ th column of  $\tilde{\Psi}^T$ , i.e.,  $\tilde{\psi}_p^T = W(\Pi_A W(\Phi^p))$ , can be regarded as an estimate for the absolute-phase discrete gradient on the graph  $\mathcal{G}_A$ . It is worth remarking that the observed (multi look) interferometric phase can however be corrupted by noise, which is taken into account by considering an additive phase noise term  $D$  [49]–[51].

Accordingly, by using (11a), we get

$$\tilde{\Psi}^T = W(W(\Pi_A \Phi) + D) = W(\Pi_A \Phi + D). \quad (13)$$

More specifically, whenever a possible spatial filtering (e.g., conventional multilooking followed by a noise-filtering step [52]) is *independently* applied to each SAR interferometric data pair, the resulting term  $D$  in (13) implies that the phase interferograms  $\tilde{\psi}^m$ , with  $m \in \{1, 2, \dots, M\}$ , are no more fully time consistent (in the sense of [25], [53], [54]). To clarify this point, we observe that, by using (13), it turns out that

$$\begin{aligned} W(\Omega_A^T \tilde{\Psi}^T) &= W(\Omega_A^T (\Pi_A \Phi + D + 2\pi Z)) \\ &= W(\Omega_A^T \Pi_A \Phi + \Omega_A^T D) = W(\Omega_A^T D) \neq 0 \end{aligned} \quad (14)$$

where we have used (11b) with  $A = \Pi_A \Phi + D$  and noted that  $\Omega_A^T Z$  is an integer matrix and  $\Omega_A^T \Pi_A = 0$  [according to (3)]. Equation (14) reads as “the wrapped discrete curl of the interferometric phase on  $\mathcal{G}_A$  (i.e., pertinent to the “temporal” domain) is generally different from zero;” it formally expresses the (modulo- $2\pi$ ) *cyclic inconsistency* of the multi-channel interferometric phase inherent to independently filtered SAR interferograms. Note also that (14) represents, within our framework, the generalization (to a wider class of discrete settings) of the “phase triangularity” condition in [53], capturing the underlying structure of the problem within a suitable

matrix formalism. With reference to the  $m$ th interferometric pair, the estimated absolute interferometric-phase gradient on the graph  $\mathcal{G}_B$  is then obtained by *wrapping the discrete gradient of (wrapped) interferometric phase field*:  $\mathbf{g}^m = W(\Pi_B \tilde{\Psi}^m)$ . Thus, by stacking the so obtained absolute phase gradient estimations, we get the  $N \times M$  matrix  $\mathbf{G} = [\mathbf{g}^1, \mathbf{g}^2, \dots, \mathbf{g}^M]$ , where

$$\mathbf{G} = W(\Pi_B \tilde{\Psi}). \quad (15)$$

Finally, by substituting (13) in (15), we obtain

$$\mathbf{G} = W(\Pi_B \tilde{\Psi}) = W(\Pi_B W([\Pi_A \Phi]^T + \mathbf{D}^T)). \quad (16)$$

From (16), by using (11b), we get

$$\begin{aligned} \mathbf{G} &= W(\Pi_B \Phi^T \Pi_A^T + \Pi_B \mathbf{D}^T + 2\pi \Pi_B \mathbf{Z}) \\ &= W(\Pi_B \Phi^T \Pi_A^T + \Pi_B \mathbf{D}^T) \end{aligned} \quad (17)$$

where in the last equality we have noted that  $\Pi_B \mathbf{Z}$  is also an integer matrix. It should be emphasized that, under the assumption  $\mathbf{D} = \mathbf{0}$ , the equality between (10) and (17) holds only up to an integer matrix multiplied by  $2\pi$ .

### C. MCh-PhU Problem as Constrained Optimization

In this section, the nonlinear inversion MCh-PhU problem is reformulated as a (nonlinear) constrained optimization problem. According to the presented general formulation, we introduce in the following the MCh-PhU problem as the solution of the following matrix equation:

$$\Pi_B \Phi^T \Pi_A^T + \Pi_B \mathbf{D}^T + 2\pi \mathbf{K} = \mathbf{G} \quad (18)$$

where the columns of  $\mathbf{G}$  represent the interferometric-phase pseudogradients estimated from the observed phase, and  $\mathbf{K}$  is an (unknown)  $N \times M$  integer matrix, whose columns represent the corresponding ( $2\pi$ -normalized) corrections to be added to the (wrapped) interferometric-phase pseudogradients in order to recover the absolute interferometric-phase discrete gradients. It is worth noting that the term “pseudogradient” is used here to emphasize that the integration of the estimated gradient is path dependent (nonconservative behavior); the term *residues* [1] is also typically used to connote the inconsistency of the estimated phase gradient. As a matter of fact, matrix equation (18) describes an *ill-posed* problem, in which the data  $\mathbf{G}$  generally do not constrain sufficiently the problem to get a unique solution. Additional suitable constraints and *a priori* assumption have, thus, to be introduced to solve the problem. First, for restoring the *cyclic consistency* (see Section II-A) of the estimated pseudogradients pertinent to the graphs  $\mathcal{G}_B$  and  $\mathcal{G}_A$ , two corresponding sets of (equality) constraints have to be enforced, respectively. More specifically, premultiplying both sides of (18) by  $\Omega_B^T$  and taking into account (3), we obtain

$$\Omega_B^T (\mathbf{G} - 2\pi \mathbf{K}) = \mathbf{0}. \quad (19)$$

Similarly, by premultiplying both sides of the transposed version of (18) by  $\Omega_A^T$  and taking also into account (3), we obtain

$$\Omega_A^T (\mathbf{G} - \Pi_B \mathbf{D}^T - 2\pi \mathbf{K})^T = \mathbf{0}. \quad (20)$$

Constraints stated by (19) imply that the columns of  $\mathbf{G} - 2\pi \mathbf{K}$  must lie in the null-space of  $\Omega_B^T$ . Since the matrix  $\Pi_B$  represents a basis to span the null space of  $\Omega_B^T$  [see (3)], we may then write  $\mathbf{G} - 2\pi \mathbf{K} = \Pi_B \mathbf{X}$ , where  $\mathbf{X}$  is a new variable. Accordingly, the corrected pseudogradients stack  $\mathbf{G} - 2\pi \mathbf{K}$  is enforced to be a stack of discrete gradients, which can thus be unambiguously integrated. Similarly, (20) implies  $[\mathbf{G} - \Pi_B \mathbf{D}^T - 2\pi \mathbf{K}]^T = \Pi_A \mathbf{Y}$ . As a result, the two sets of constraints, which is stated by (19) and (20), guarantee that the solution of the problem is effective in preserving the *cyclic consistency* (curl-free) property of the corrected gradients pertaining to the graphs  $\mathcal{G}_B$  and  $\mathcal{G}_A$ , respectively. As a matter of fact, the solution of (18) cannot be determined by using the two sets of constraints (19) and (20), only; thus, the inverse problem must be first regularized [55]. The minimum-norm methods search for a global solution that minimizes a generalized error-norm associated with an optimality criterion, therefore incorporating prior information about the behavior of the solution [1]. Accordingly, we resort to a regularization approach using  $l_1$ -norm minimization in weighted version, as a specific case of  $l_p$ -norm general formulation. Formally, the MCh-PhU problem may be then formulated as a constrained optimization problem for the field of integer corrections as follows:

$$\hat{\mathbf{K}} = \arg \min_{\mathbf{K} \in \mathbb{Z}^{N \times M}} \|\mathbf{K}\|_{1, \mathbf{C}} \quad (21)$$

subject to

$$\begin{cases} \Omega_A^T \mathbf{K}^T = \Omega_A^T [\mathbf{G} - \Pi_B \mathbf{D}^T]^T (2\pi)^{-1} \\ \Omega_B^T \mathbf{K} = \Omega_B^T \mathbf{G} (2\pi)^{-1} \end{cases} \quad (22)$$

wherein

$$\|\mathbf{K}\|_{1, \mathbf{C}} = \sum_{m=1}^M \sum_{n=1}^N c_{nm} |k_{nm}| \quad (23)$$

represents the weighted  $l_1$ -norm [56] of the matrix  $\mathbf{K}$ ,  $\mathbf{C} = [c_{nm}]_{N \times M}$  denotes a suitable weighting matrix, and  $\mathbb{Z}$  indicates the field of integer numbers. As far as the existence of an integer solution for (21) and (22) is concerned, it should be noted that the considerations at the end of Section II-A apply. Since the first matrix equation in (22) includes a generally nonnull (unwanted) term  $\Omega_A^T \mathbf{D}$ , its fulfillment deserves further discussion. Although the evaluation of  $W(\Omega_A^T \mathbf{D})$  can be obtained according to (14), a full estimation for  $\Omega_A^T \mathbf{D}$  is, however, generally difficult to obtain. Note that we are interested in finding out proper integer corrections  $\mathbf{K}$ , so that  $\Omega_A^T \mathbf{K}^T$  also results in an integer matrix. Therefore, according to [25], we can instead consider the set of constraints, i.e.,

$$\Omega_A^T \mathbf{K}^T = \lfloor \Omega_A^T \mathbf{G}^T (2\pi)^{-1} \rfloor \quad (24)$$

which can be strictly fulfilled in terms of integer corrections and where  $\lfloor \cdot \rfloor$  symbolizes the nearest integer function.



The solution of the optimization problem (21) and (22) is also referred to as the minimum weighted discontinuity solution (in a weighted  $l_1$ -norm sense) [1], [14]. As a matter of fact, finding the global minimum of the problem (21) and (22) for an arbitrary pair of graphs is, in general, a difficult task. A suboptimal strategy aimed at solving the problem (21) and (22) consists in adopting a two-stage approach. This is, in particular, the solution strategy implemented through the EMCF technique, in which the edge structure of each considered graph is usually defined via a *Delaunay* triangulation in the Euclidean plane, to take advantage from efficient solvers for minimum cost flow (MCF) problems [57], as briefly summarized in the following Section III for the sake of completeness.

As a final remark, for our purposes, it is important to note that, assuming for the sake of simplicity  $\mathbf{D} = \mathbf{0}$ , the equality constraints in (22) can be also explicitly rewritten as

$$\begin{cases} \Omega_A^T [\mathbf{k}_1^T, \dots, \mathbf{k}_N^T] = \Omega_A^T [\mathbf{g}_1^T, \dots, \mathbf{g}_N^T] (2\pi)^{-1} \\ \Omega_B^T [\mathbf{k}_1^T, \dots, \mathbf{k}_M^T] = \Omega_B^T [\mathbf{g}_1^T, \dots, \mathbf{g}_M^T] (2\pi)^{-1}. \end{cases} \quad (25)$$

Finally, note also that, in the classical single-channel case ( $M = 1$ ), the formulation (21) and (22) reduces to

$$\hat{\mathbf{k}} = \arg \min_{\mathbf{k} \in \mathbb{Z}^N} \|\mathbf{k}\|_{1,\mathbf{c}} \quad (26)$$

subject to

$$\Omega_B^T \mathbf{k} = \Omega_B^T \mathbf{g} (2\pi)^{-1} \quad (27)$$

where

$$\|\mathbf{k}\|_{1,\mathbf{c}} = \sum_{n=1}^N c_n |k_n| \quad (28)$$

is the weighted  $l_1$ -norm [56] of the  $N$ -dimensional vector  $\mathbf{k}$ , and  $\mathbf{c} = [c_1, \dots, c_N]^T$  is a suitable weighting vector.

### III. RATIONALE OF THE EMCF ALGORITHM

We describe, in this section, the sequential version of the EMCF algorithm [25]. EMCF provides an extension to the multichannel context of the single-channel MCF approach [7], [58], [59] (valid for planar graphs). Although EMCF has widely been discussed in several papers with reference to its implications in applications perspective [26], [60]–[64], for our purposes, the algorithm structure deserves a more systematic and detailed description. Accordingly, in the following, the EMCF algorithm is reviewed and cast into an algebraic framework, by adopting the suitable operators of the discrete calculus. This formal description constitutes a propedeutic step toward the devising of a relevant parallel computational model, in order to gain insight into the computational structure of the sequential version, and understand the computational cost of the different processing steps. The remainder of this section consists of four parts. Section III-A preliminarily introduces the model-based approach for the interferometric phase in a vectorial notation. Section III-B and C detail the two-stage and overall optimization strategies of EMCF, respectively. Finally,

Section III-D addresses a numerical solution via the MCF-based approach.

#### A. Model-Based Approach

Let us refer to the vectors  $\mathbf{t} = [t_1, \dots, t_Q]^T$  and  $\mathbf{b}_\perp = [b_{\perp 1}, \dots, b_{\perp Q}]^T$  collecting, respectively, the *time* and *perpendicular baseline* of the considered SAR acquisitions, whose entries are expressed with reference to the master acquisition. Thus, with reference to the graph  $\mathcal{G}_A$ , the discrete gradients  $\Pi_A \mathbf{b}_\perp$  and  $\Pi_A \mathbf{t}$  represent  $M$ -dimensional vectors, whose  $m$ th elements are, respectively, the *perpendicular* and *temporal baseline* (i.e., the difference between the acquisition times) relative to the  $m$ th interferometric SAR data pair [25]. The interferometric phase includes contributions due to the *residual topography*  $\Delta z$  and surface *displacement mean velocity*  $\Delta v$  pertinent to a certain edge of the graph  $\mathcal{G}_B$ , respectively [25]. These contributions can be calculated via a simple linear phase model [25]. By collecting the modeled interferometric phase for the considered stack of  $M$  interferometric SAR data pairs, a compact vector-valued functional description  $(\Delta z, \Delta v) \in \mathbb{R}^2 \rightarrow \Theta \in \mathbb{R}^M$  is introduced:

$$\Theta(\Delta z, \Delta v) = \frac{4\pi}{\lambda} \left( \frac{\Delta z}{r \sin \vartheta} \Pi_A \mathbf{b}_\perp + \Delta v \Pi_A \mathbf{t} \right) \quad (29)$$

where  $\vartheta$  is the look angle,  $r$  is the sensor-to-pixel distance, and  $\lambda$  is the central wavelength of the transmitted signal. Note also that, because of the spatially correlated behavior of the atmospheric artifacts [17], we have assumed in (29) that they do not contribute to the model and the nonlinear deformation signal component. It should be noted that  $\Theta(\Delta z, \Delta v)$  describes a discrete gradient on the graph  $\mathcal{G}_A$ , by construction [note that  $\Omega_A^T \Theta(\Delta z, \Delta v) = \mathbf{0}$ , according to (3)] The rationale of the model-based approach originally proposed in [25] was then to subtract the modeled contribution from the interferometric phase to reduce the relevant phase variation, thus facilitating the subsequent PhU operations. Accordingly,  $\forall n \in \{1, 2, \dots, N\}$ , we get

$$\tilde{\mathbf{g}}_n^T(\Delta z, \Delta v) = \Theta(\Delta z, \Delta v) + W(\mathbf{g}_n^T - \Theta(\Delta z, \Delta v)). \quad (30)$$

Note that  $\Delta z$  and  $\Delta v$ , which are related to the observed Earth's surface deformation phenomena, are unknown quantities.

#### B. Two-Stage Scheme

The EMCF procedure is composed of two stages for solving the problem formulated in (21) and (22) [25]. Indeed, a natural strategy for solving the problem (21) and (22), since it has a very large number of constraints, is based on relaxation. Accordingly, the EMCF procedure starts by solving a relaxed version of (21) and (22) that ignores some constraints. Specifically, the *first-stage* addresses the constraints enforced by the first equation in (22) only. The relaxed version of the original optimization problem (21) and (22) can be then directly formulated as a set of  $N$  (independent) constrained minimization problems [see also (25)], with  $n$  denoting the  $n$ th edge of the



graph  $\mathcal{G}_B$ , so that the unknown matrix  $\mathbf{K}$  is accordingly handled row by row, i.e.,

$$\begin{aligned} \forall n \in \{1, 2, \dots, N\}, \\ \hat{\mathbf{k}}_n^{(s)} = \arg \min_{\mathbf{k} \in \mathbb{Z}^M} \|\mathbf{k}\|_1 \quad (31) \\ (\Delta z, \Delta v) \in \Omega^{(s)} \end{aligned}$$

subject to

$$\Omega_A^T \mathbf{k}^T = [\Omega_A^T \tilde{\mathbf{g}}_n^T (\Delta z, \Delta v) (2\pi)^{-1}] \quad (32)$$

where  $\Delta z$  and  $\Delta v$  are treated as free continuous variables,  $\Omega^{(s)}$  is a suitable bounded domain constraining the feasibility region for the searched  $\Delta z$  and  $\Delta v$  unknowns,  $\mathbb{Z}^M$  is the set of  $M$ -tuples on integers, and the *nonlinear* objective function is the  $l_1$ -norm of the  $M$ -dimensional row vector  $\mathbf{k} = [k_1, \dots, k_M]$ . We emphasize that hereinafter the superscript  $(s)$  denotes a quantity pertinent to a specific searching domain  $\Omega^{(s)}$ . It is worth highlighting that, for a fixed  $n$ , the subproblem (31) and (32) describes a *mixed-integer nonlinear programming* (MINLP) problem involving the integer variables represented by the vector  $\mathbf{k} \in \mathbb{Z}^M$ , and the couple of continuous variables  $(\Delta z, \Delta v)$  pertain to the  $n$ th edge of the graph  $\mathcal{G}_B$ . As a result, by collecting the  $M$ -dimensional integer row vectors  $\hat{\mathbf{k}}_n^{(s)}$  estimated  $\forall n \in \{1, 2, \dots, N\}$  via (31) and (32), for a prescribed searching domain  $\Omega^{(s)}$ , we obtain the following  $N \times M$  integer matrix:

$$\hat{\mathbf{K}}^{(s)} = \begin{bmatrix} \hat{\mathbf{k}}_1^{(s)} \\ \hat{\mathbf{k}}_2^{(s)} \\ \vdots \\ \hat{\mathbf{k}}_N^{(s)} \end{bmatrix}. \quad (33)$$

The estimated  $\hat{\mathbf{K}}^{(s)}$  is then applied at first to correct the original matrix  $\tilde{\mathbf{G}} = [\tilde{\mathbf{g}}_1^T, \dots, \tilde{\mathbf{g}}_N^T]^T$ , thus obtaining

$$\hat{\mathbf{F}}^{(s)} = \tilde{\mathbf{G}} - 2\pi \hat{\mathbf{K}}^{(s)}. \quad (34)$$

The  $N \times M$  matrix  $\hat{\mathbf{F}}^{(s)} = [\hat{\mathbf{f}}_1^{(s)}, \dots, \hat{\mathbf{f}}_M^{(s)}]$  estimated via the previous stage is subsequently used as a starting point for the second-stage operation, which only addresses the constraints enforced by the second equation in (22). Accordingly, the *second-stage* operation is formulated as a set of  $M$  constrained minimization problems with integer variables (INLP), with  $m$  denoting the  $m$ th edge of the graph  $\mathcal{G}_A$ , so that the current integer matrix describing the (unknown) residual normalized correction, here denoted  $\mathbf{H}$ , can be now handled column by column, i.e.,

$$\begin{aligned} \forall m \in \{1, 2, \dots, M\} \\ \hat{\mathbf{h}}^m = \arg \min_{\mathbf{h} \in \mathbb{Z}^N} \|\mathbf{h}\|_{1, \mathbf{w}^{(s)}} \quad (35) \end{aligned}$$

subject to

$$\Omega_B^T \mathbf{h} = \Omega_B^T \hat{\mathbf{f}}^m (2\pi)^{-1}. \quad (36)$$

Note that the objective functions to be minimized is the weighted  $l_1$ -norm of the  $N$ -dimensional integer column vector  $\mathbf{h} = [h_1, \dots, h_N]^T$ , and  $\mathbf{w}^{(p)} = [w_1^{(s)}, \dots, w_N^{(s)}]^T$  is a proper weighting vector that gives a confidence measure of the correctness of the phase estimation carried out in the first (temporal) stage of the PhU scheme [25]. As a result, by collecting the estimated  $N$ -dimensional integer column vectors  $\hat{\mathbf{h}}^m$  obtained  $\forall m \in \{1, 2, \dots, M\}$  via (35), (36), we obtain the  $N \times M$  integer matrix, i.e.,

$$\hat{\mathbf{H}}^{(s)} = [\hat{\mathbf{h}}^1, \dots, \hat{\mathbf{h}}^M]. \quad (37)$$

By taking into account the estimated integer-valued correction matrices (37) and (34), we retrieve an estimate of the unwrapped phase gradients on the graph  $\mathcal{G}_B$  for the whole stack of interferograms, i.e.,

$$\hat{\mathbf{S}}^{(s)} = \hat{\mathbf{F}}^{(s)} - 2\pi \hat{\mathbf{H}}^{(s)} = \tilde{\mathbf{G}} - 2\pi \hat{\mathbf{K}}^{(s)} - 2\pi \hat{\mathbf{H}}^{(s)}. \quad (38)$$

Finally, each column vector  $\hat{\mathbf{s}}^m$  of the matrix (38) is integrated over a generic path covering the entire 2-D phase spatial grid, thus obtaining an estimate of the unwrapped phase stack as follows:

$$\hat{\Psi}^{(s)} = [\hat{\psi}^1, \hat{\psi}^2, \dots, \hat{\psi}^M] = \begin{bmatrix} \hat{\psi}_1^{(s)} \\ \hat{\psi}_2^{(s)} \\ \vdots \\ \hat{\psi}_P^{(s)} \end{bmatrix} \quad (39)$$

where the  $P$ -dimensional column vector  $\hat{\psi}^m$  represents the unwrapped phase pertinent to the  $m$ th interferometric pair, with  $m \in \{1, 2, \dots, M\}$ , and  $P$  denoting the number of pixels of interest.

### C. Rationale of the Adopted Optimization Strategy

Each stage of the approach discussed in Section III-B may provide a germane feasible solution; however, the overall two-stage solution may not be the optimal one. Indeed, due to the highly nonlinear character of the considered problem and the presence of many potential local minima, the choice of a suitable optimization strategy is thus important because it drives the quality of the estimated solution. Accordingly, instead of directly estimating  $\hat{\mathbf{k}}_n$ , by allowing  $(\Delta z, \Delta v)$  to vary in a prescribed searching 2-D space  $\Omega$ , successive candidate solutions, i.e.,  $\hat{\mathbf{k}}_n^{(s)}$  with  $s \in \{1, 2, \dots, S\}$ , are considered over a sequence of nested searching subdomains of the original configuration space  $\Omega \subset \mathbb{R}^2$ , as fully detailed in the following. A nested set of bounded subdomains  $\Omega^{(S)} \subset \dots \subset \Omega^{(2)} \subset \Omega^{(1)} \subseteq \Omega$  is primarily constructed. Accordingly, for each subdomain  $\Omega^{(s)}$ ,  $N$  column vectors  $\hat{\mathbf{k}}_n^{(s)}$  with  $n = 1, 2, \dots, N$ , are estimated via (31) and (32), and then  $M$  row vectors  $\hat{\mathbf{h}}^m$  with  $m = 1, 2, \dots, M$  are estimated via (35) and (36). Eventually, the unwrapped interferometric phase (for each interferometric pair) is reconstructed via path integration, thus obtaining  $\hat{\Psi}^{(s)} = [\hat{\psi}^1, \dots, \hat{\psi}^M]$ . Hence, the estimation scheme, in which

the objective function is minimized over nested subdomains of the original searching space  $\Omega$ , considers a sequence of  $S$  two-stage optimization problems. Accordingly, this scheme permits a “constrained” exploration of the configuration space over the sequence of nested subdomains. The key idea behind this approach is to exploit a proper consistency estimation of the reconstructed phase in combining the obtained solutions  $\hat{\psi}^{m(s)}$ ,  $s \in \{1, 2, \dots, S\}$ . An estimation of the pertinent consistency is provided via the *temporal coherence* [25]. Accordingly, for each  $m$ th interferometric pair, a combined solution  $\hat{\psi}^m$  is obtained as

$$\hat{\psi}^m = \sum_{s=1}^S \bar{\Gamma}^{(s)} \circ \hat{\psi}^{m(s)} \quad (40)$$

where  $\circ$  denotes the *Hadamard* product, and  $\bar{\Gamma}^{(s)} = [\bar{\Gamma}_1^{(s)}, \bar{\Gamma}_2^{(s)}, \dots, \bar{\Gamma}_P^{(s)}]^T$  is the  $P$ -dimensional normalized temporal coherence vector (see Appendix A). The purpose of the weighted average in (40) is to de-emphasize certain solution points affected by low temporal coherence, in order to mitigate the undue influence of local errors inherent in the individual predictions. Therefore, this strategy is naturally robust with respect to local minima, due to the incorporation of the estimated temporal coherence into the multidomain optimization scheme.

We finally obtain the estimated  $m$ th unwrapped interferometric phase field, according to [13], as

$$\tilde{\psi}^m + 2\pi \left\lfloor \frac{\hat{\psi}^m - \tilde{\psi}^m}{2\pi} \right\rfloor \quad (41)$$

where  $\tilde{\psi}^m$  represents the original (wrapped) interferometric phase field, with  $m \in \{1, 2, \dots, M\}$ .

#### D. Minimum-Cost Network-Flow-Based Approach

In this section, we focus on the optimization problem implementation structure, due to its impact on the sequential algorithm pattern (see Fig. 2) and relevant parallel design strategy (as discussed in Section IV).

It is worth highlighting that EMCF approach reduces the MCh-PhU problem to subproblems that, from an implementation perspective, can be numerically solved by resorting essentially to the same basic approach (MCF). Specifically, the basic subproblems (35) and (36), and (31) and (32) are formally identical or can be led back to the one in (26) and (27), respectively. Accordingly, we primary focus attention on the concrete solution of the basic INLP problem (26) and (27), which can be generally rewritten in the following form:

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{Z}^N} \|\beta\|_{1,c} \quad \text{subject to} \quad \Omega^T \beta = \mathbf{d}. \quad (42)$$

Indeed, as recognized in [7], the problem stated in (42) can also be viewed as an instance of *minimum cost network flow* (MCF) problem, for which consolidated solvers exist [57], [65], [66].

By resorting to the notion of *duality* for abstract graphs [42], [45], the same result can be established within the adopted alge-

braic framework; the formal derivation in terms of differential operators of the discrete calculus is provided in Appendix B.

We believe that the solution strategy for the MINLP subproblem in (31) and (32) deserves further considerations. Generally speaking, there is a very wide variety of deterministic methods to solve *mixed-integer linear programming* problems. However, only a few of them are applicable to MINLP problems. As a matter of fact, a large number of heuristics have been developed to solve practical MINLP problems [55]. More specifically, it is worth noting that the subproblem (31) and (32), with respect to the integer variables  $\mathbf{k} \in \mathbb{Z}^M$  only, can also be regarded as an instance of the problem (42). It is thus clear that the resulting optimization problem, which is obtained from the subproblem (31) and (32) by fixing the pair of variables  $(\Delta z, \Delta v)$ , can be profitably solved by using an MCF-based approach. Accordingly, an approach inspired to the grid search method [55] might conveniently be used to find out an approximate minimum for the MINLP subproblem (31), (32). More precisely, a prescribed grid is first set up in the entire domain, namely  $\Omega^{(1)}$ , and all the nested subdomains  $\Omega^{(1)} \supseteq \Omega^{(2)} \dots \supseteq \Omega^{(S)}$  inherit it, thus creating a relevant discretized 2-D solution space. Then, for a certain  $s \in \{1, 2, \dots, S\}$ , numerically solving pertinent MCF problem at all the grid points  $(\Delta z_i, \Delta v_j) \in \Omega^{(s)}$  and finding the grid point  $(\Delta z_{\hat{i}}, \Delta v_{\hat{j}})$  that corresponds to the lowest objective function value are accomplished through an exhaustive search over the discretized solution subdomain pertaining  $\Omega^{(s)}$ .

Furthermore, as described in Section III-C, the subproblem in (31) and (32) has to be evaluated over  $S$  nested domains. Thus, since a prescribed grid is assumed, it is advantageous to perform searching over the nested subdomains  $\Omega^{(1)} \supseteq \Omega^{(2)} \dots \supseteq \Omega^{(S)}$  in a sequential way starting from  $s = 1$ , in order to avoid redundant evaluations. Indeed, it should be noted that once a pertinent solution point  $(\Delta z_{\hat{i}}, \Delta v_{\hat{j}})$  has been obtained via exploration of a certain discretized solution subdomain pertaining  $\Omega^{(t)}$ , every successive evaluation pertinent to a subdomain  $\Omega^{(s)} \subseteq \Omega^{(t)}$  with  $s \in \{t+1, \dots, S\}$  and  $s > t$  can be avoided whenever  $(\Delta z_{\hat{i}}, \Delta v_{\hat{j}}) \in \Omega^{(t)} \cap \Omega^{(s)}$ .

The resulting computational structure of sequential implementation of the EMCF algorithm is schematically shown in Fig. 2.

#### IV. PARALLEL COMPUTATIONAL MODEL

In this section, the emphasis is placed on the parallel formulation of the EMCF algorithm discussed in Section III. This is done by stressing the methodological and practical aspects leading to a *dual-level* parallel algorithm structure, in which the parallelism is hierarchically implemented at two different levels.

Generally speaking, developing a parallel model for a certain application basically consists of two parts. Primarily, the formulation of a specific parallelization strategy, which critically depends on the original algorithm structure, must be devised; then, a pertinent implementation has to be addressed (see Section V). As far as the former issue is concerned, it is crucial to decompose the problem at hand into elements (subproblems) that can be executed concurrently. Thus, recasting the original

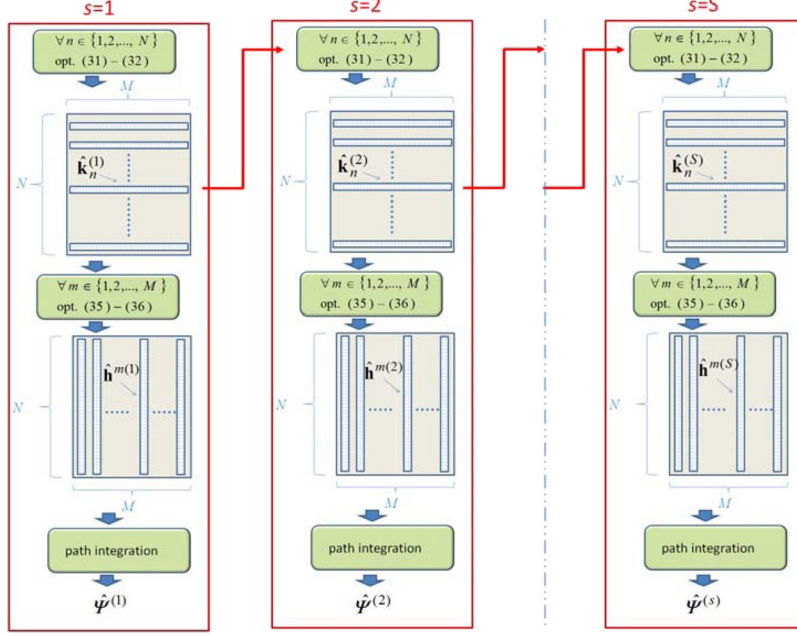


Fig. 2. Sequential version of EMCF algorithm: schematic representation.

algorithm into a form (decomposition pattern) [32] that exposes available concurrency inherent into the problem is a primary concern. Flexibility, simplicity, and efficiency/scalability are the three main guidelines that should dictate the adopted decomposition pattern strategy [32]. In particular, as far as efficiency (i.e., effective use of the available computing resources) of a parallel scheme is concerned, the main factors regard load-balancing and overhead (mainly associated with scheduling, synchronization, and communication) minimization. It is worth highlighting that communication is a source of significant overhead particularly on *distributed-memory* platforms; conversely, synchronization is the main sources of overhead in *shared-memory* machines [30]–[35]. Therefore, we first address the supporting hardware to which our parallel model is oriented (see Section IV-A), and then, we discuss the adopted task-decomposition strategy, also focusing on the relevant dependency analysis (see Section IV-B). Section IV-C and D detail the different levels of the attained parallel structure.

#### A. Current Architectures and Dual-Level Parallelism

Preliminarily, target computational infrastructures are briefly considered since parallel algorithms and parallel architectures are closely tied together, and different parallel models for parallel computing can be considered according to the way in which processors are connected. Parallel computing architecture can be generally categorized in two main classes. In *shared-memory* multiprocessors, common memory can be accessed by all processors. On the contrary, in *distributed-memory* multiprocessors, the local memories are only accessible through the respective processors [37], [40], [41]. As a matter of fact, commonly available parallel architectures are based on hybrid combination of shared-memory and distributed-memory systems. Indeed, high-performance computing (HPC) infrastructures are

essentially clusters, which are typically based on *distributed-memory* multinode architecture for scalable performance, in which each node incorporate *shared-memory* multicore processors. To properly take advantage from the hierarchical parallelism offered by this emerging class of high performance computer architectures and successfully achieve the performance that these architectures can offer, we consider the design of a suitable parallel computational model based on a dual-level scheme for the considered application. Accordingly, our parallel scheme, since it is aptly oriented to such target architectures, encompasses both coarse-grain distributed-memory parallelism and fine-grain shared-memory parallelism. Hence, the associate parallel prototype implements both message-passing, for communication between nodes, and multithreading, within a multiprocessor node, approaches to parallel programming (hybrid programming model).

#### B. Dependence Analysis and Task Decomposition

Here, the challenge is structuring the parallel algorithm to take advantages from the potential concurrency inherent to the problem. For such a purpose, first of all, the dependencies intrinsic to the specific problem, which hinders direct concurrency exploitation, have to be carefully identified and correctly managed. Contextually, the formulation of a dual-level parallel model is then obtained via the decomposition of the computational problem at hand into a hierarchical collection of components (tasks/subtasks) that can be executed concurrently, thus expressing the nested parallelism that can be implemented at different levels.

First, dependency analysis is addressed. By inspection of the algorithm pattern in Fig. 2, data-flow dependencies between different macroblocks (denoted by  $s=1, \dots, S$ ), which impose a required order on their execution, can be easily recognized.

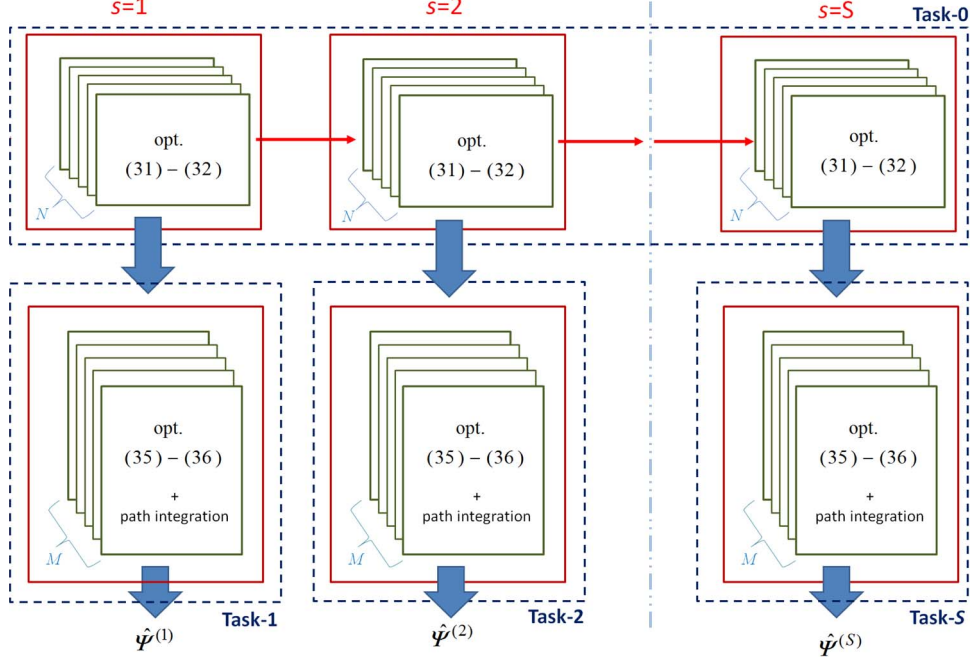


Fig. 3. Parallel model: task-decomposition schematic representation.

Indeed, these constraints arise from the order in which the exploration of the discretized solution subdomains must be performed, according to the optimization strategy discussed in Section III-D. Due to these ordering constraints between computationally significant parts of the EMCF algorithm (see red arrows in Fig. 2), the problem at first glance appears to be inherently sequential. Indeed, it is not possible to directly obtain independent tasks that can be carried out in parallel, although the algorithm incorporates some regular structures. As a matter of fact, the existing data-flow dependencies cannot be directly eliminated; thus, they have a major impact on the parallel design.

To overcome such a limitation, we rely on a proper functional decoupling of the computational problem that must be properly managed to permit safe concurrent execution, as discussed in the following. The aim is to identify, by splitting the iteration into serial and parallel portions, a suitable decomposition of the problem into subproblems that in part can be solved concurrently. Accordingly, we rearrange the relevant algorithm structure, to obtain  $S$  disjoint computations pertinent to EMCF (spatial-PhU) *second-stage* (each one including  $M$  optimization problems (35), (36) and relevant path integration), as schematically depicted in the scheme of Fig. 3: We identify the first part (task-0), comprising the sequences of  $S$  subparts [each one including  $N$  optimization problems (31) and (32)], and further parts (task-1, task-2,  $\dots$ , task- $S$ ), each one involving the solution of  $M$  optimization problems (35), (36) pertaining to a different  $s \in \{1, 2, \dots, S\}$ . Accordingly, task-0 and the set (task-1, task-2,  $\dots$ , task- $S$ ) can be computed serially in relation to each other, so enabling  $S$  disjoint computations (task-1,  $\dots$ , task- $S$ ).

As it can be noted, the approach we use follows a practice usually adopted to cope with the encountered kind of data-flow dependency (see for instance [32], [40], and [67]). Fig. 3 out-

lines the resulting decomposition scheme and pertinent data flow.

### C. Distributed-Memory Parallelism

The adopted decomposition suitably breaks up the original application in a number of tasks (according to the pattern schematically indicated in Fig. 3), thus enabling relevant concurrent computations of (task-1, task-2,  $\dots$ , task- $S$ ) in a coarse-grain fashion at highest level. We now discuss diverse perspectives supporting the adopted task-decomposition.

First, some computational remarks are in order to gain insight into the computational cost of the several parts of the parallel algorithm. It is important to note that the computational cost of the MCF algorithm grows more than linearly with the prescribed number of edges  $J$  of the considered graph, following the behavior described by the function  $f(J)$  [66]. Concerning the tasks (task-1, task-2,  $\dots$ , task- $S$ ), the associated computational cost of each of these grows as  $M \cdot f(N)$ ; thus, the overall cost associated with them essentially increases as  $S \cdot M \cdot f(N)$ . On the contrary, the computational cost of each subpart of the task-0 grows as  $N \cdot f(M)$ ; thus, the cost associated with task-0 increases as  $S \cdot N \cdot f(M)$ . As a matter of fact, for common scenarios of interest,  $S$ ,  $M$ , and  $N$  are typically on the order of tens, tens if not hundreds, and thousands if not millions, respectively. Since it turns out that  $N \gg M$  (because the number of pixels of interest is typically several orders of magnitude greater than the number of SAR acquisitions), it is clear that the computational cost associated with task-0 results significantly reduced with respect to the one of the remaining tasks, especially for high coherent scenarios and high-resolution sensors ( $N$  is on the order of millions). Second, the resulting arrangement provides for tasks (task-1, task-2,  $\dots$ , task- $S$ ) that are loosely coupled. This offers a high

degree of coarse-grain parallelism that can be implemented with extremely reduced communication (data transfer between nodes) and synchronization overhead (as detailed in Section V-A). Note also that this concern can be crucial if the parallel model is implemented on the emerging *cloud computing* platforms [68], [69]. These considerations suggest that the adopted scheme can be therefore successfully implemented on distributed-memory parallel architectures. It should be also noted that, although further partitioning of the task-0 might be considered at this level, the arising amount of data to communicate between computing nodes could be indeed considerable, with a consequent efficiency detriment.

As a further strength of our approach, we emphasize that this scheme is also effective since it easily enables a further (second) level of parallelism, as detailed in the following. How the identified tasks can be implemented in the form of separate software processes, to be allocated to different processors, is addressed in Section V-A.

#### D. Shared-Memory Parallelism

Once we have restructured the algorithm in the previously discussed form, we are therefore in position to profitably exploit the potential concurrency inherent to the inner structure of the identified tasks. Indeed, the inner structure of each relevant task is iterative in nature, and straightforwardly, it breaks down into a collection of independent subtasks, thus suggesting a second level of parallelization. Specifically, for each  $s \in \{1, 2, \dots, S\}$ , the task- $s$  embraces a set of  $M$  iterations (see Fig. 3), so that a fine-grain subtask decomposition is naturally obtained by associating each iteration on a subtask. Similar considerations apply for the set of  $S$  blocks included within the task-0: Each block comprises a set of  $N$  iterations (see Fig. 3). Note that, since the iterations are completely independent and numerous, the corresponding subtasks are relatively simple to manage concurrently. Indeed, due to the large number of interferometric pairs ( $M$  is typically on the order of tens if not hundreds), there will be many subtasks associated with the set of  $M$  iterations, thus making the granularity of the decomposition suitable for shared-memory system. Similar considerations apply as well for all the  $S$  blocks included in the task-0 because  $N$  is typically on the order of thousands if not millions. As far as the data associated with each subtask is concerned, we emphasize that each one essentially needs information about the corresponding estimated (pseudo) gradient. The adopted read-only model for shared input data is then effective on a shared-memory system since each subtask can read distinct data from the entire (pseudo) gradient-stack data structure as needed. As a result, in this case, a row-based (or column-based) data decomposition is implied by the considered subtask decomposition [32]. In addition, the output of subtasks can be defined in terms of concurrent updates of distinct segment (rows or columns) of a 2-D array. A final practical issue to be considered concerns how the subtasks map on the different computing elements in the form of separate software threads, and it is discussed in Section V-B according to the resultant load-balancing and scalability.

### V. STRUCTURED PARALLELISM IMPLEMENTATION

According to the dual-level parallel model proposed in Section IV, the parallel application results abstractly divided into a hierarchical arrangement of tasks and subtasks, thus specifying how the parallelism can be implemented at two different levels. Proper *load balancing* is a critical concern for efficient parallel computing [32], [37], [40], [41]. Accordingly, we now focus on the relevant prototype implementation, with special emphasis on the distribution of the computational load among the computing units (processors/processor cores), thus focusing on pertinent mapping and task scheduling for software implementation. Mapping parallelism on architectural resources however implies some insight into specific programming models and tools. Accordingly, the implementation of the parallel scheme in the form of separate software *processes* and *threads* is addressed in subsections A and B/C, respectively.

#### A. Process-Level Parallelism

To exploit distributed-memory architectures, we utilize a *message-passing* model [40], [41] for implementing the top level of parallelism. First of all, we point out that the independent tasks (task-1, task-2,  $\dots$ , task- $S$ ) involve approximately the same amount of computation. Therefore,  $S$  tasks are divided in chunks that are assigned to the participating processors for concurrent execution. Accordingly, the inherent coarse-grain parallelism is implemented by distributing the chunks to participating processors in a *round-robin* fashion [32], [40], [41]. We emphasize that the considered round-robin scheme is a *static* cyclic scheduling strategy (the tasks are distributed among the processors prior to the execution with an arrangement configurable at runtime) with minimum additional communication and synchronization cost; its adoption is effective in our case, due to the uniform computational requirements. In addition, the approach we adopt is simple and easy to implement on distributed-memory parallel architectures. Thus, efficient intra-node message passing functionality has been straightforward implemented via simple scripting language (instead of the common approach based on MPI [34]). As a result, the implemented scheduling approach for process-level parallelism, which is aimed at maximizing locality and minimize communication, is advantageous since it provides a good compromise between load balancing and parallel efficiency, thus permitting to achieve high performance (as discussed in Section VI).

#### B. Thread-Level Parallelism

To exploit shared-memory multicore processors (at node level), we adopt a *fork-join* model of parallel execution [40], [41] for implementing the lower level of parallelism.

We concentrate on the parallel regions enclosing the groups of subtasks identified in Section IV-D, respectively. Accordingly, the inherent fine-grain parallelism is implemented, at a thread level, by distributing the subtasks of each group to the engaged computing elements (processor cores), as discussed in the following.

Preliminarily, it is worth emphasizing that fine-grain parallelism generally suffers from the drawback of frequent thread creation, destruction, and associated synchronization [32], [40], [67]. We stress that our scheme has the important advantage of avoiding such detrimental overhead since it essentially considers the most external loops (associated with each identified group of subtasks) as the primary source of parallelism. In addition, no data that are modified within the relevant loops are shared between the threads; therefore, they can run with no need for further synchronization (see also Section V-C).

Specific considerations concerning the scheduling strategy to be adopted for load balancing are now in order. For such a purpose, we focus on a single parallel region enclosing a certain group of subtasks since similar considerations apply as well for the other groups of subtasks. Remarkably, we point out that, since the amount of work required by the MCF solver critically depends on the specific instance of the considered optimization problem, subsequently, the execution time of different subtasks can vary in an irregular way and is difficult to predict in advance. Accordingly, the adoption of a *static* scheduling strategy cannot be appropriate in this case since different threads will get vastly different and unpredictable workloads and finish the loop at different times, thus leading to load imbalance. On the contrary, *guided* and *dynamic* scheduling strategies, allowing a more flexible workload distribution, can be adopted to counteract load imbalance. However, since the coordination of assignment is needed in this case, requests for iterations incur some additional synchronization cost. Note also that static scheduling has lower overhead, but it cannot compensate for load imbalances by shifting more iterations to less heavily loaded threads [40], [41]. In all cases, subtasks are assigned to threads in contiguous ranges called chunks; the number of subtasks that a chunk contains is the chunk size. More specifically, the idea underlying guided scheduling strategy is to initially schedule large chunks and, later, to use smaller chunks to smoothen the unevenness of the execution times of the initial larger chunks. On the contrary, dynamic scheduling handles the chunks with first-come, first-serve scheme [30]–[33].

Open multiprocessing (OPENMP) [67] is here employed for practical implementation of thread-level parallelism. OPENMP is a portable (cross-platform) parallel programming interface for shared-memory machines, which is a well-established standard for HPC and is widely adopted to obtain thread parallelism for scientific computing applications [32], [40], [41]. OPENMP consists of a set of language extensions implemented as compiler directive, thus providing mechanisms for explicit specification of the thread-based parallelism. Such directives allow expressing the parallelism in the code implementation in a highly structured way [67].

Performance analysis supporting our design strategy is conducted in Section VI, also exploring and discussing the efficiency achievable with different scheduling strategies.

### C. Thread-Safe Version of the RELAX-IV Routine

The adopted MCF-based solution strategy leverages on the utilization of efficient network-flow solvers, as thoroughly discussed in Section III-D. More details on practical implemen-

TABLE II  
REFERENCE COMPUTATIONAL PLATFORM CHARACTERISTICS

<i>Operative System</i>	GNU/Linux (kernel version 2.6.32)
<i>Processor</i>	8-core CPU - 2.6 GHz Intel Xeon E5-2670
<i>Number of nodes</i>	16
<i>Memory Size per node</i>	384 GB of RAM
<i>Storage</i>	14 disks 4 TB, SAS 6Gb/s
<i>Interconnection</i>	56 Gb/s InfiniBand

tation aspects concerning the concurrent evaluations of MCF within a shared-memory environment are briefly provided. Specifically, in order to solve MCF problems, we make use of the available RELAX-IV routine [57], [66]. Its FORTRAN implementation, unfortunately, makes intensive use of common blocks as workspace. Thus, a serious data consistency problem clearly arises when multiple threads simultaneously execute different instances of the RELAX-IV routine since, during parallel execution, all threads share the common workspace. To overcome such a limitation, one possibility is to provide some form of synchronization. As a matter of fact, this practice can be unacceptably inefficient in terms of parallel performance, due to the associated synchronization overheads. Therefore, to maintain data coherence across all concurrent threads, we instead replace these common areas with private areas for each thread. Accordingly, data stored in the workspace are local to each thread, and the program initializes the workspace prior to each use. In this way, our thread-safe version of RELAX-IV guarantees consistent concurrent execution of different instances of the MCF algorithm with thread parallelism.

## VI. EXPERIMENTAL RESULTS

In this section, the quantitative evaluation of the computational efficiency of the implemented parallel prototype is addressed to experimentally demonstrate the effectiveness of the proposed parallel EMCF approach.

### A. Computational Platform

The experimental analysis has been carried out by using a multinode cluster, consisting of 16 computing nodes, each one equipped with two CPU (eight-core 2.6-GHz Intel Xeon E5-2670) and 384 GB of RAM. The employed computational platform has a storage system including fourteen 4-TB disks and Serial Attached SCSI (SAS) interface disk drives (in a RAID-5 configuration), and each drive is capable of sustaining 6 GB/s. In addition, a 56-Gb/s InfiniBand interconnection is employed, and a Network File System (NFS) is implemented for file sharing (see Table II).

### B. Case Study Description

The Naples Bay area, a volcanic and densely urbanized district in Southern Italy (including the Campi Flegrei active caldera, the Vesuvius volcano and the city of Naples), has been selected as a representative test area for the experimental investigation. First, an ASAR/ENVISAT data set composed by 64 acquisitions, collected over ascending orbits (track 36, frame 2781) from November 2002 to September 2010, has



TABLE III  
REFERENCE CASE STUDY: SAR DATA SET PARAMETERS

Operative area size		$100 \times 100 \text{ Km}^2$
Temporal span		Nov. 2002 – Sept. 2010
Single-look image size		$18000 \times 5000$
Multi-look image size		$900 \times 1250$
Number of SAR acquisitions	$Q$	64
Number of interferograms	$M$	175
Number of edges of $\mathcal{G}_B$	$N$	340009
Number of selected pixels	$P$	113382

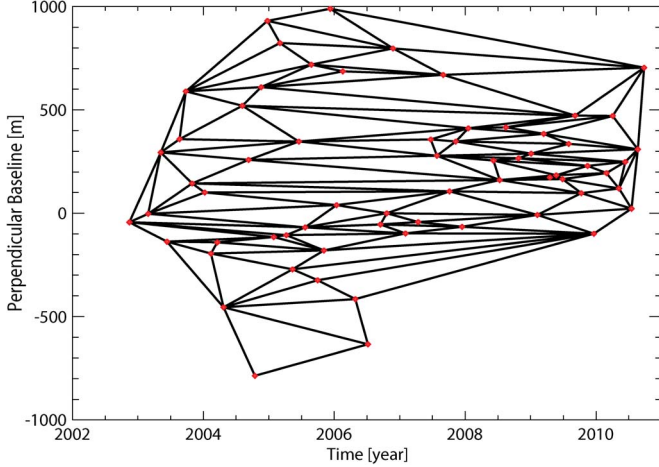


Fig. 4.  $\mathcal{G}_A$  graph obtained via Delaunay triangulation in the Euclidean (temporal baseline, perpendicular baseline) plane, whose nodes represent the set of  $Q$  SAR acquisitions on the investigated area.

been employed. In particular, 175 differential interferograms, characterized by a (perpendicular) baseline value smaller than 400 m and a maximum temporal separation of about 1500 days, have been generated. The main reference characteristic parameters for the considered data sets are summarized in Table III. The graphs  $\mathcal{G}_A$  and  $\mathcal{G}_B$  pertinent to this case study are obtained via Delaunay triangulation and depicted in Figs. 4 and 5, respectively.

### C. Simulated Results

In order to better clarify the effectiveness of the proposed optimization strategy (presented in Section III-C), further investigations have also been carried out on simulated data. To achieve this task, assuming the same topological setting discussed in Section VI-B, we have suitably synthesized a stack of SAR interferograms, with the aim of quantifying the benefit of the multidomain-based strategy in terms of solution accuracy. More details on relevant analysis are provided in Appendix C.

### D. Sequential Equivalence

We briefly discuss the correctness of the results obtained via our parallel prototype. It is important to note that floating-point arithmetic is nonassociative, so that the order in which algebraic manipulations are carried out can slightly affect the resulting numerical values [32], [40]. Although our parallel version involves floating-point operations performed in different order, with respect to the corresponding sequential version, so that

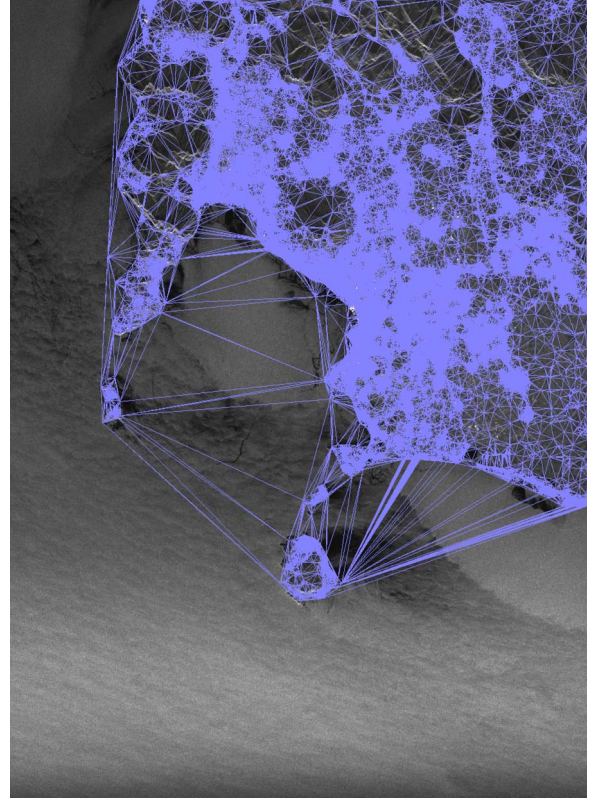


Fig. 5.  $\mathcal{G}_B$  graph obtained via Delaunay triangulation in the Euclidean (azimuth, range) plane, whose nodes represent the set of coherent pixels, superimposed on an amplitude SAR image of the investigated area.

slight changes can occur in the associated intermediate results; however, our prototype gives identical final results when executed on one processor or many (sequential equivalence). This is due to both the specific parallel solution strategy and the nature of the problem. Concerning the latter, note also that it essentially involves the determination of integer quantities (gradient corrections), which remain indeed unaffected by errors related to floating-point arithmetic. The *sequential equivalence* has been also experimentally verified.

### E. Performance Metrics and Modeling

In order to quantitatively appreciate the benefit of parallelism achievable via the proposed solution, we suitably resort to performance measures that are typically adopted. Specifically, we preliminary introduce the notions of *speedup* and *efficiency* [30]–[33]. The *speedup*  $S = S(\text{NE})$  of a parallel program with *parallel execution time*  $T_{\text{NE}}$  is defined as

$$S(\text{NE}) = \frac{T_1}{T_{\text{NE}}} \quad (43)$$

where NE is the number of computing elements (processors or processor cores) used to solve a problem, and  $T_1$  is the (*best*) execution time of the sequential implementation to solve the same problem [30]–[33]. We emphasize that, in the following, we make use of the most rigorous definition of speedup, which sets the sequential reference time  $T_1$  to the execution time of the *best* sequential algorithm corresponding to the parallel algorithm under study [32]. It should be noted that the upper



bound for the speedup is given by  $S(\text{NE}) \leq \text{NE}$ . Accordingly, due to unavoidable presence of inefficiencies, the parallel linear speedup ( $S(\text{NE}) = \text{NE}$ ) is achievable only in ideal conditions. An alternative measure for the performance of a parallel program is the *efficiency*, defined as

$$\varepsilon = \frac{S(\text{NE})}{\text{NE}}. \quad (44)$$

It should be noted that an ideal speedup corresponds to unitary efficiency. A theoretical limit on the performance benefit of increasing the number of computing elements is expressed by the well-known *Amdahl's law* [30]–[35], whose generalized formulation is the following:

$$S(\text{NE}) = \frac{1}{O(\text{NE}) + (1 - f_p) + \frac{f_p}{\beta \text{NE}}} \quad (45)$$

where the  $f_p$  is the fraction of a program that can be executed in parallel,  $O = O(\text{NE})$  is the system overhead, and  $\beta \in [\text{NE}^{-1}, 1]$  is the *average imbalance factor*. Notice when the workload is perfectly balanced, we get  $\beta = 1$ ; conversely,  $\beta = 1/\text{NE}$  when the entire workload is carried out by a single computing element. Note that the performance model in the form (45) is amenable to the analysis of both multiprocessor and multicore systems.

#### F. Performance Analysis

In this section, we conduct a performance analysis aimed at investigating the effectiveness of the proposed solution at the different levels of parallelism. The conducted analysis is also aimed at evaluate pertinent scalability by assuming a fixed problem (described in Section VI-B) and is carried out on the computational infrastructure described in Section VI-A.

Specifically, we first focus on the analysis of the process-level parallelism in which a *static* scheduling approach is used; subsequently, the thread-level parallelism is considered. Detailed performance analysis of the process-level parallelism inherent to (task-1, task-2, ..., task- $S$ ) is now concerned. Relevant speedup has been experimentally carried out on up to 16 processors and is shown (blue line/diamonds) in Fig. 6 as a function of the number of engaged processors; in addition, the ideal linear speedup (dashed line) is also depicted as reference. As a matter of fact, high performances are achieved in exploiting the available distributed-memory parallelism, as it is evident from the graph. A further insight on the obtained results can be gained by specializing the performance model (45). Indeed, for the case under study, it is reasonable to assume  $f_p \cong 1$  and  $O(\text{NE}) \cong 0$ . Accordingly, the speedup predicted via (45) (with an estimated  $\beta \cong 0.93$ ) is also depicted (red line/circles) in Fig. 6. We point out that the speedup predicted via this simple performance model results to be in a good agreement with the experimental results. It is clear that the experimental results can be essentially explained in terms of a residual load unbalancing, which hampers achieving ideal (linear) performance. In this regard, it should be noted that residual load imbalance among processors is closely connected with the coarse granularity implied by the top level of the parallel scheme. Note also that, in this case, the granularity of the parallel workload depends on

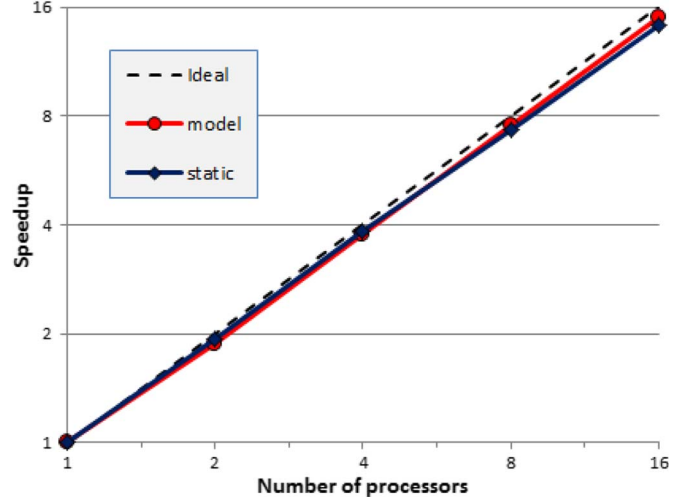


Fig. 6. Speedup as a function of the number of engaged processors (blue/diamonds). The ideal linear speedup (dashed line) and the speedup predicted by the considered model (red/circles) are also depicted.

$S$  (i.e., the number of searching nested subdomains, discussed in Section III-C). Therefore, we have demonstrated the effectiveness of the compromise, between load balancing and parallel efficiency, which underpins the top level of the proposed parallel scheme, and the appropriateness of the associated static scheduling strategy (see also Section V-A).

We now investigate in detail the attained performance pertinent to the shared-memory parallelism at the node level. First of all, we explore the performances achievable with different scheduling scenarios for load balancing at thread level. Specifically, static scheduling (SS), guided scheduling (GS), and dynamic scheduling (DS) have been implemented for performance comparison. For such a purpose, in the following, we focus on a single parallel region enclosing a generic group of subtasks associated with a generic task- $s$ , with  $s \in \{1, 2, \dots, S\}$  (see Section V-B) since similar considerations apply, *mutatis mutandis*, for the groups of subtasks included in the task-0. We emphasize, in this case, that the granularity of the workload depends on the problem size (specifically on  $M$ , i.e., the number of edges of the graph  $\mathcal{G}_A$ ). In Fig. 7(a), speedup versus the number of engaged processor cores is depicted for three scheduling strategies: SS (blue line/diamonds), GS (yellow line/squares) and DS (red line/triangles). The ideal linear speedup (dashed line) is also shown as reference. As shown in Fig. 7(a), the adoption of the SS scheme, which however implies a very low overhead, results in a notably performance degradation as the number of threads/cores increases. It should be noted that the SS strategy is not able to counteract the load imbalance resulting from the (actual) highly uneven completion times of the relevant subtasks, thus wasting performance opportunity. As it is notable from Fig. 7(a), the GS strategy partially succeeds in accommodating this kind of workload variance, thus implying a certain unbalancing. As a result, the experimental analysis we have carried out confirms (see Section V-B), in our case, that DS strategy outperforms the other strategies, thus permitting to minimize the overall execution time and also showing remarkable scalability. Therefore, the adopting DS strategy is appropriate in our case. The benefits

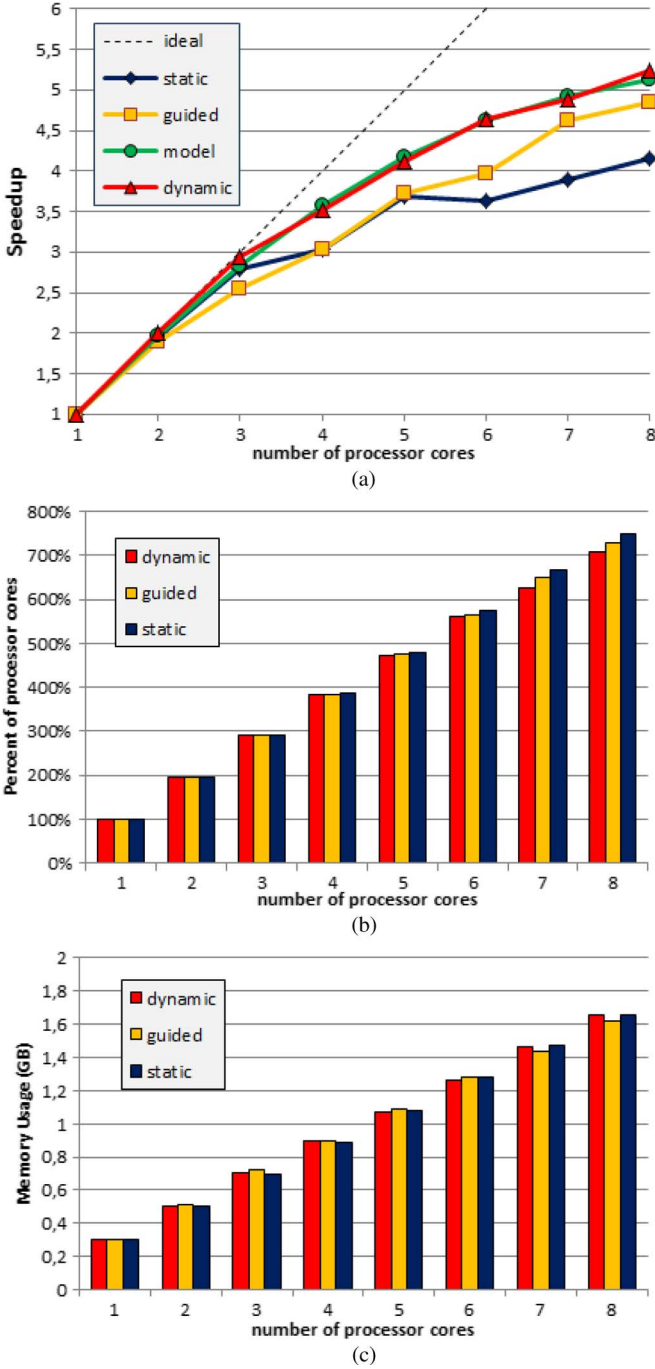


Fig. 7. Multithreading performance. (a) Speedup, (b) processor cores usage (in percentage), and (c) maximum resident memory size (in gigabytes), versus the number of engaged processor cores, obtained for different scheduling strategies: static (blue), guided (yellow), and dynamic (red). The ideal linear speedup (dashed line) and the speedup predicted by the considered model (green) are also depicted.

of the load balancing achievable with the DS strategy are however paid in terms of greater associated overhead, which hampers achieving ideal (linear) performance [see Fig. 7(a)]. To better investigate the residual inefficiency inherently DS strategy, we employ (45) as a performance model. For the case under study, it is reasonable to assume  $f_p \cong 1$  and  $\beta \cong 1$ . In addition, since, in a shared-memory context, the system overhead, associated with thread management overhead (creation/destruction and synchronization) and memory band-

width contention among multiple threads, generally increases with the number of threads and hence physical processor cores, we employ a simple linear model for the system overhead, i.e.,  $O(NE) = \Delta(NE - 1)$ , in (45). As a result, the speedup predicted by the considered performance model (obtained with an estimated  $\Delta \cong 1\%$ ), whose behavior is also depicted in Fig. 7(a) (green line/circles), shows a good agreement with the experimentally observed performance (DS).

Furthermore, for the three different scheduling strategies, the percentage of usage of the processor cores and the maximum resident memory size (in gigabytes), versus the number of engaged processor cores, have also been depicted in Fig. 7(b) and (c), respectively. In particular, the memory usage pattern can be a critical aspect, in terms of memory requirement, and deserves to be carefully taken into consideration when multithreading is concerned. In this regard, note also that, in this case, the memory requirement per thread essentially increases as  $N$  (i.e., the number of edges of the graph  $\mathcal{G}_B$ ). Similar results have been obtained for the groups of subtasks included in the task-0, and they are not shown due to space limitations.

As a result, we have demonstrated that our parallel solution permits to make an efficient use of the available computing resources, showing significant speedup and scalability at different (process and thread) levels. In addition, further insights into the factors of residual inefficiency, at the different levels of parallelism, have been provided by employing a simple model. For the sake of completeness, we also remark that the sequential execution time (which however is strictly tied to the specific hardware reported in Table II) was, for the case under study, of about 30 h.

## VII. FINAL REMARKS

The aim of this paper is twofold. First, we have formally established a rigorous framework in which the MCh-PhU problem can be formulated in terms of meaningful mathematical objects. From a mathematical point of view, *discrete calculus* offers an appropriate playground in which relevant differential operators are properly defined on a discrete setting and are intrinsically related to the *topological* structures associated with the discrete spaces upon which they act. Its adoption is not only a matter of formalism but allow us to overcome the conceptual (and topological) inconsistencies arising from the improper use of conventional differential operators on a discrete setting. Accordingly, the MCh-PhU problem formulation has consistently been cast within this theoretical framework, with the following remarkable advantages. Pertinent formulation has systematically been phrased in a matrix form, through the application of the discrete operators properties and their relationship, thus reflecting the topological structure of the MCh-PhU problem, whose representation is made mathematically apparent. Interestingly, our formalism have also permitted to analytically investigate the impact on MCh-PhU of *independently filtered* SAR interferograms. Therefore, the resulting *cyclic inconsistency* of the multichannel interferometric phase has been formally characterized, in a general way, in terms of discrete-curl operator, thus including the “phase triangularity” of [53], [54] as a particular case. In addition, within our framework,

the applicability of well-known powerful MCF (numerical optimization) solvers formally emerges as a consequence of the intimate connection between the *planarity* and *duality* graph-theoretical concepts. Finally, this formal framework is also useful for a better understanding of the computational costs associated with the different parts of the EMCF algorithm.

Second, we have proposed a novel parallel computational model for the specific EMCF solution strategy, to face the time-consuming nature of MCh-PhU problem, which hampers its large-scale application in a reasonable time frame. It is worth mentioning that structuring this computationally intensive application has been obtained by methodological targeting to emerging (hierarchical) multiprocessing platforms. Accordingly, a distinctive feature of the proposed dual-level parallel model resides in its ability to achieve good performance in exploiting the available degrees of parallelism of the current HPC platforms; this includes both shared-memory and distributed-memory architectures. Several important aspects (dependence analysis, task-partitioning, scheduling, etc.) related to our design strategy have extensively been discussed, also emphasizing the methodological approach and its implications. Practical implementation (in terms of software processes and threads) of the proposed parallel model has been also addressed, clarifying the specific embraced models and tools.

Performance evaluation relevant to the obtained parallel EMCF prototype solution has been carried out, thus quantifying the benefit of parallelism at different levels. We have obtained significant speedup with a subsequent dramatic reduction of the relevant execution time. In addition, we have shown that our solution permits to maintaining efficiency while increasing the number of engaged computing elements (that is the scalability goal in scientific computing). As a result, we have experimentally demonstrated the validity of our approach.

As far as added acquisitions are concerned, different approaches can be applied (see for instance [70]). Nonetheless, it is worth highlighting that the speedup achievable with our parallel solution makes no more so critical the issue of added acquisitions.

As far as the basic assumptions inherent to the EMCF method are concerned, we finally emphasize that the assumption on the linearity is made for the ground deformation difference across a certain edge (connecting two spatial points) of the graph  $\mathcal{G}_B$  (pertinent to the spatial domain), rather than for the ground deformation evaluated as node variable of the same graph. It is clear that the nearer the graph nodes, the more appropriate is this assumption; in particular, this is true even in the presence of large spatial discontinuities, as successfully demonstrated in [60]–[64], [71]–[73] for different application scenarios, including seismotectonic and volcanic phenomena.

#### APPENDIX A

##### NORMALIZED TEMPORAL COHERENCE

The *temporal coherence* factor  $\Gamma_p^{(s)} \in [0, 1]$ , pertaining to the  $p$ th pixel of interest and  $s$ th subdomain, is defined as

$$\Gamma_p^{(s)} = \frac{1}{M} \left| \sum_{m=1}^M e^{j r_p^{m(s)}} \right| \quad (\text{A1})$$

where  $j$  is the imaginary unit,  $M$  is the number of interferometric pairs, and  $r_p^{m(s)}$  is the *residual error* (relevant to the  $m$ th interferometric pair) defined in [25]. The temporal coherence is assumed as a reliability indicator of the reconstructed interferometric phase: the success of the interferometric-phase retrieval inherent to the  $p$ th pixel (and  $s$ th searching subdomain) is then manifested in high value of  $\Gamma_p^{(s)}$ ; on the contrary, low values of the relevant temporal coherence correspond to poorly unwrapped data [25]. Let us consider the *normalized temporal coherence* factors, i.e.,

$$\bar{\Gamma}_p^{(s)} = \frac{\Gamma_p^{(s)}}{\sum_{s=1}^S \Gamma_p^{(s)}}. \quad (\text{A2})$$

Therefore, the normalized-temporal-coherence vector is the  $P$ -dimensional vector defined as

$$\bar{\Gamma}^{(s)} = [\bar{\Gamma}_1^{(s)}, \dots, \bar{\Gamma}_P^{(s)}]^T. \quad (\text{A3})$$

We emphasize that  $\bar{\Gamma}^{(s)}, \forall s \in \{1, 2, \dots, S\}$  provides information about the temporal consistency of the corresponding  $s$ th solution  $\hat{\psi}^{m(s)}$  (note that  $m$  identifies the generic interferometric pair), thus offering a criterion to combine  $S$  different solutions.

#### APPENDIX B

##### MCF, DUALITY, AND PLANARITY

We now formally establish within the adopted algebraic framework the main result of [7]. First of all, it is worth highlighting a fundamental topological characterization for the planarity property: A graph is planar if and only if it has an abstract dual [45]. Let us consider a planar graph  $\mathcal{G}$  characterized by  $Q$  nodes,  $M$  edges, and  $R$  faces. In the case of the *planar* graph  $\mathcal{G}$ , one can identify a *dual* graph  $\mathcal{G}^*$  with node-face duality defined by  $\bar{Q} = R + 1$ ,  $\bar{R} = Q - 1$ , and  $\bar{M} = M$ . It can be demonstrated that further differential operator and function duality exist [42]. Hence, if  $\alpha$  and  $\beta$  are the solution of equations  $\mathbf{I}^T \alpha = \mathbf{b}$  and  $\mathbf{O}^T \beta = \mathbf{d}$  on the planar graph  $\mathcal{G}$ , respectively, then another solution on the dual graph  $\mathcal{G}^*$  is obtained, via the following set of formal substitutions:

$$\begin{cases} \bar{\alpha} = \beta; & \bar{\mathbf{b}} = \mathbf{d}; & \bar{\mathbf{I}}^T = \mathbf{O}^T \\ \bar{\beta} = \alpha; & \bar{\mathbf{d}} = \mathbf{b}; & \bar{\mathbf{O}}^T = \mathbf{I}^T. \end{cases} \quad (\text{B1})$$

Thus, by merely substituting different variables into the expression (42), we obtain (by duality) the following problem:

$$\hat{\bar{\alpha}} = \arg \min_{\bar{\alpha} \in \mathbb{Z}^{\bar{M}}} \|\bar{\alpha}\|_{1,c} \text{ s. t. } \bar{\mathbf{I}}^T \bar{\alpha} = \bar{\mathbf{b}}. \quad (\text{B2})$$

Therefore, the solution of the problem in (B.2) can be directly related to the solution of the original problem (42). Finally, for a given  $\bar{\alpha} = [\bar{\alpha}_1, \dots, \bar{\alpha}_{\bar{M}}]^T$ , we consider the decomposition  $\bar{\alpha} = \bar{\alpha}^+ - \bar{\alpha}^-$ , with  $\bar{\alpha}^+ = [\max(0, \bar{\alpha}_1), \dots, \max(0, \bar{\alpha}_{\bar{M}})]^T$  and  $\bar{\alpha}^- = -[\min(0, \bar{\alpha}_1), \dots, \min(0, \bar{\alpha}_{\bar{M}})]^T$ , so that the problem (B.2) can be reformulated in the following matrix form:

$$(\hat{\bar{\alpha}}^+, \hat{\bar{\alpha}}^-) = \arg \min_{\hat{\bar{\alpha}}^+, \hat{\bar{\alpha}}^- \in \mathbb{N}_0^{\bar{M}}} \mathbf{c}^T (\hat{\bar{\alpha}}^+ + \hat{\bar{\alpha}}^-) \quad (\text{B3})$$

$$\text{s. t. } \bar{\mathbf{I}}^T (\bar{\alpha}^+ - \bar{\alpha}^-) = \bar{\mathbf{b}} \quad (\text{B4})$$

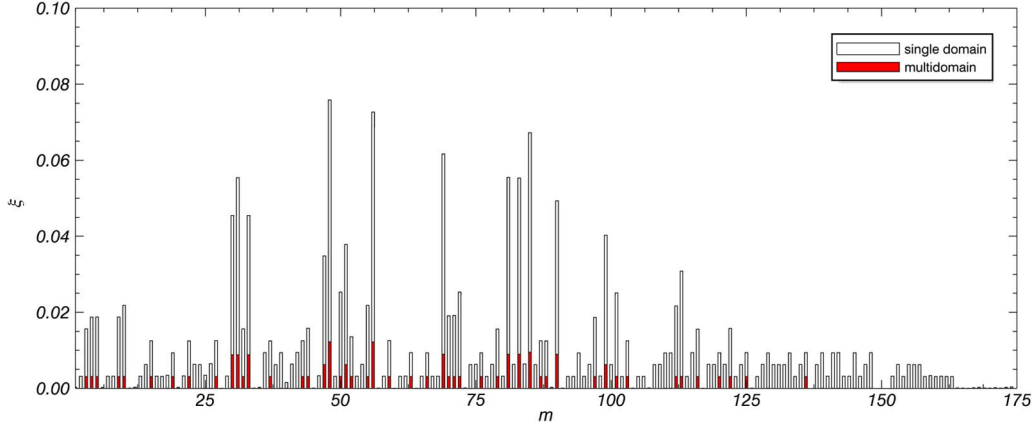


Fig. 8. Normalized error  $\xi(m)$  versus the interferogram number ( $m$ ) pertinent to both single domain (white) and multidomain (red) computations.

which defines a special class of *linear* programs referred to as MCF problems [65]. The field of nonnegative integer numbers is denoted by  $\mathbb{N}_0$ . It should be also noted that  $\|\bar{\alpha}\|_{1,c} = c^T(\bar{\alpha}^+ + \bar{\alpha}^-)$ . This interesting derivation that we have provided, formally clarifying how the solution for MCF problem (B.3) and (B.4) can be used for solving the original problem (42), gives a clear insight into the intimate connection between graph-theoretic concepts of planarity and duality, which plays a key role for the MCF-based solution-strategy application.

### APPENDIX C SIMULATION EXPERIMENTS

To corroborate the effectiveness of the adopted optimization scheme, which has been detailed in Section III-C, simulated experiments are presented. For the sake of simplicity, the simulated experiments have been performed by considering the same topological setting (see Figs. 4 and 5) adopted for the case study discussed in Section VI-B. The simulated interferograms, namely  $\psi^m$  with  $m \in \{1, 2, \dots, M\}$ , have been synthesized, following the guidelines of [25], as

$$\psi^m = \frac{4\pi}{\lambda} \left( \frac{\Delta b_{\perp m}}{r \sin \vartheta} \mathbf{z}^{(\text{sim})} + \Delta t_m \mathbf{v}^{(\text{sim})} + \frac{1}{2} (\Delta t_m)^2 \mathbf{a}^{(\text{sim})} \right) \quad (\text{C1})$$

where  $\Delta b_{\perp m}$  and  $\Delta t_m$  are the perpendicular and temporal baselines related to the  $m$ th SAR data pair, respectively, and the  $P$ -dimensional vectors  $\mathbf{z}^{(\text{sim})}$ ,  $\mathbf{v}^{(\text{sim})}$ , and  $\mathbf{a}^{(\text{sim})}$  describe the node variables (on  $\mathcal{G}_B$ ) pertinent to the simulated topography, velocity, and acceleration, respectively. Specifically,  $\mathbf{z}^{(\text{sim})}$  has been obtained by computing the high-frequency spatial components of the digital elevation model of the case study area, and a (spatial) Gaussian-shaped model has been adopted for both  $\mathbf{v}^{(\text{sim})}$  and  $\mathbf{a}^{(\text{sim})}$ . Note that the model parameters (for velocity and acceleration: maximum amplitude of 50 cm/year and 15 cm/year<sup>2</sup>, and standard deviation of 8 cm/year and 0.8 cm/year<sup>2</sup>) we have considered are representative of a worst case involving an intense Earth's deformation field. Starting from the wrapped synthesized interferograms  $\tilde{\psi}^m = W(\psi^m)$ , the EMCF solution  $\hat{\psi}^m$  has been evaluated by considering both

the single-domain ( $\Omega^{(1)}$ ) and multidomain (see Section III-C) cases. We subsequently evaluate,  $\forall m \in \{1, 2, \dots, M\}$ , the discrepancy between the estimated (unwrapped)  $\hat{\psi}^m$  and the original (synthesized) interferometric phase  $\psi^m$  [obtained via (C.1)], by introducing a suitable normalized error  $\xi = \xi(m)$  defined as

$$\xi(m) = \frac{1}{P} \left\| \frac{\hat{\psi}^m - \psi^m}{2\pi} \right\|_1 \quad (\text{C2})$$

where  $P$  is the number of pixels of interest. The normalized error, defined by (C.2), has been evaluated for both the solutions obtained via the single-domain ( $\Omega^{(1)}$ ) and multidomain (see Section III-C) strategy, respectively: It is depicted in Fig. 8 as a function of interferogram index  $m \in \{1, 2, \dots, M\}$ . As it can be noted from the figure, the single-domain solution implies an error that ( $\forall m$ ) is much greater than the corresponding one pertinent to multidomain case. Therefore, the simple analysis we have conducted demonstrates the usefulness of the adopted multi-domain optimization scheme.

### ACKNOWLEDGMENT

The authors would like to thank S. Guarino, F. Parisi, and M.C. Rasulo for their valuable technical support; the European Space Agency for providing the ENVISAT ASAR data; and the University of Delft, The Netherlands, for the related precise orbits. The digital elevation model of the investigated zones were acquired through the SRTM archive.

### REFERENCES

- [1] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*. New York, NY, USA: Wiley, 1998.
- [2] R. M. Goldstein and H. A. Zebker, "Mappings small elevation changes over large areas: Differential radar interferometry," *J. Geophys. Res.*, vol. 94, no. B7, pp. 9183–9191, Jul. 1989.
- [3] A. Hooper and H. Zebker, "Phase unwrapping in three dimensions with applications to InSAR time series," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 24, no. 9, pp. 2737–2747, Aug. 2007.
- [4] R. Bürgmann, P. A. Rosen, and E. J. Fielding, "Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation," *Annu. Rev. Earth Planet. Sci.*, vol. 28, pp. 169–209, May 2000.

- [5] D. Massonnet and K. L. Feigl, "Radar Interferometry and its application to changes in the Earth's surface," *Rev. Geophys.*, vol. 36, pp. 441–500, 1998.
- [6] G. Fornaro, A. Pauciuolo and D. Reale, "A null-space method for the phase unwrapping of multitemporal SAR interferometric stacks," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2323–2334, Jun. 2011.
- [7] 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.
- [8] R. Gens, "Two-dimensional phase unwrapping for radar interferometry: Developments and new challenges," *Int. J. Remote Sens.*, vol. 24, no. 4, pp. 703–710, 2003.
- [9] 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.
- [10] C. W. Chen and H. A. Zebker, "Network approaches to 2-D phase unwrapping: Intractability and two new algorithms," *J. Opt. Soc. Amer.*, vol. 17, no. 3, pp. 401–414, Mar. 2000.
- [11] G. F. Carballo and P. W. Fieguth, "Hierarchical network flow phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 8, pp. 1695–1708, Aug. 2002.
- [12] K. Zhang *et al.*, "Phase unwrapping for very large interferometric data sets," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 4048–4061, Oct. 2011.
- [13] W. Xu and I. Cumming, "A region-growing algorithm for InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, no. 1, pp. 124–134, Jan. 1999.
- [14] T. J. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 14, no. 10, pp. 2692–2701, 1997.
- [15] G. Fornaro, G. Franceschetti, and R. Lanari, "Interferometric SAR phase unwrapping using Green's formulation," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 3, pp. 720,727, May 1996.
- [16] G. Fornaro, G. Franceschetti, R. Lanari, D. Rossi, and M. Tesaro, "Interferometric SAR phase unwrapping using the finite element method," *Proc. Inst. Elect. Eng.—Radar, Sonar Navigat.*, vol. 144, no. 5, pp. 266,274, Oct. 1997.
- [17] A. Ferretti, C. Prati, and F. Rocca, "Permanent scatterers in SAR interferometry," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 1, pp. 8–20, Jan. 2001.
- [18] C. Werner, U. Wegmüller, T. Strozzi, and A. Wiesmann, "Interferometric point target analysis for deformation mapping," in *Proc. IGARSS*, Toulouse, France, Jul. 21–25, 2003, vol. 7, pp. 4362–4364.
- [19] A. Hooper, H. Zebker, P. Segall, and B. M. Kampes, "A new method for measuring deformation on volcanoes and other natural terrains using InSAR persistent scatterers," *Geophys. Res. Lett.*, vol. 31, no. 23, pp. 1–5, Dec. 2004.
- [20] B. M. Kampes, *Radar Interferometry: Persistent Scatterer Technique*. Dordrecht, The Netherlands: Springer-Verlag, 2006.
- [21] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2375–2383, Nov. 2002.
- [22] O. Mora, J. J. Mallorquí, and A. Broquetas, "Linear and nonlinear terrain deformation maps from a reduced set of interferometric SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 10, pp. 2243–2253, Oct. 2003.
- [23] M. Crosetto, B. Crippa, and E. Biescas, "Early detection and in-depth analysis of deformation phenomena by radar interferometry," *Eng. Geol.*, vol. 79, no. 1/2, pp. 81–91, Jun. 2005.
- [24] S. Usai, "A least squares database approach for SAR interferometric data," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 4, pp. 753–760, Apr. 2003.
- [25] A. Pepe and R. Lanari, "On the extension of the minimum cost flow algorithm for phase unwrapping of multitemporal differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2374–2383, Sep. 2006.
- [26] A. Pepe, L. D. Euillades, M. Manunta, and R. Lanari, "New advances of the extended minimum cost flow phase unwrapping algorithm for SBAS-DInSAR analysis at full spatial resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 4062–4079, Oct. 2011.
- [27] A. P. Shanker and H. Zebker, "Edgelist phase unwrapping algorithm for time series InSAR analysis," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 27, no. 3, pp. 605–612, Mar. 2010.
- [28] M. Costantini *et al.*, "A general formulation for robust integration of finite differences and phase unwrapping on sparse multidimensional domains," in *Proc. Fringe*, Frascati, Italy, Dec. 2009, pp. 758–768.
- [29] R. Bamler and P. Hartl, "Synthetic aperture radar interferometry," *Inverse Problems*, vol. 14, no. 4, pp. R1–R54, 1998.
- [30] I. Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Reading, MA, USA: Addison-Wesley, 1995.
- [31] J. J. Dongarra *et al.*, *Sourcebook of Parallel Computing*. San Francisco, CA, USA: Morgan Kaufman, 2003.
- [32] T. G. Mattson, *et al.*, *Patterns for Parallel Programming*. Boston, MA, USA: Addison-Wesley, 2005.
- [33] P. S. Pacheco, *An Introduction to Parallel Programming*. Burlington, MA, USA: Morgan Kaufmann, 2011.
- [34] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge, MA, USA: MIT Press, 1999.
- [35] S. G. Akl, *The Design and Analysis of Parallel Algorithms*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.
- [36] Special issue on high performance computing in earth observation and remote sensing, *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 503–507, Sep. 2011.
- [37] G. Hager and G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*. Boca Raton, FL, USA: CRC Press, 2010.
- [38] F. Casu *et al.*, "SBAS-DInSAR parallel processing for deformation time series computation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 8, pp. 3285–3296, Aug. 2014.
- [39] P. Imperatore, *et al.*, "Scalable performance analysis of the parallel SBAS-DInSAR algorithm," in *Proc. IEEE IGARSS*, Québec City, QC, Canada, Jul. 2014, pp. 350–353.
- [40] H. El-Rewini and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*. New York, NY, USA: Wiley, 2005.
- [41] F. Gebali, *Algorithms and Parallel Computing*. Hoboken, NJ, USA: Wiley, 2011.
- [42] L. J. Grady and J. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. New York, NY, USA: Springer-Verlag, 2010.
- [43] N. Biggs, *Algebraic Graph Theory*. Cambridge, U.K.: Cambridge Univ. Press, 1994.
- [44] C. Berge, *Graphs and Hypergraphs*. Amsterdam, The Netherlands: North-Holland Publishing Co., 1973.
- [45] R. Diestel, *Graph Theory*. New York, NY, USA: Springer-Verlag, 2000.
- [46] L. Grady, "Minimal surfaces extend shortest path segmentation methods to 3D," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 32, vol. 2, pp. 321–334, Feb. 2010.
- [47] P. A. Rosen *et al.*, "Aperture radar interferometry," *Proc. IEEE*, vol. 88, no. 3, pp. 333–381, Mar. 2000.
- [48] H. A. Zebker and J. Villasenor, "Decorrelation in interferometric radar echoes," *IEEE Trans. Geosci. Remote Sens.*, vol. 30, no. 5, pp. 950–959, Sep. 1992.
- [49] C. H. Gierull, "Statistical analysis of multilook SAR interferograms for CFAR detection of ground moving targets," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 4, pp. 691–701, Apr. 2004.
- [50] C. Lopez-Martinez and E. Pottier, "On the extension of multidimensional speckle noise model from single-look to multilook SAR image," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 2, pp. 305–320, Feb. 2007.
- [51] J. S. Lee, K. W. Hopple, S. A. Mango, and A. R. Miller, "Intensity and phase statistics of multilook polarimetric interferometric SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 5, pp. 1017–1028, Sep. 1994.
- [52] R. M. Goldstein and C. L. Werner, "Radar interferogram filtering for geophysical applications," *Geophys. Res. Lett.*, vol. 25, pp. 4035–4038, 1998.
- [53] A. Pepe, Y. Yang, M. Manzo, and R. Lanari, "Improved EMCF-SBAS processing chain based on advanced techniques for the noise-filtering and selection of small baseline multi-look DInSAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4394–4417, Aug. 2015.
- [54] A. Ferretti *et al.*, "A new algorithm for processing interferometric data-stacks: SqueeSAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 9, pp. 3460–3470, Sep. 2011.
- [55] S. S. Rao, *Engineering Optimization: Theory and Practice*, 4th ed. New York, NY, USA: Wiley, 2009.
- [56] K. Yosida, *Functional Analysis*. Berlin, Germany: Springer-Verlag, 1980.
- [57] D. Bertsekas and P. Tseng, "The relax codes for linear minimum cost network flow problems," *Ann. Oper. Res.*, vol. 13, pp. 125–190, 1988.



- [58] M. Costantini and P. A. Rosen, "A generalized phase unwrapping approach for sparse data," in *Proc. IGARSS*, Hamburg, Germany, 1999, pp. 267–269.
- [59] M. Costantini, A. Iodice, L. Magnapane, and L. Pietranera, "Monitoring terrain movements by means of sparse SAR differential interferometric measurements," in *Proc. IGARSS*, Honolulu, HI, USA, 2000, pp. 3225–3227.
- [60] E. Trasatti *et al.*, "The 2004–2006 uplift episode at Campi Flegrei caldera (Italy): Constraints from SBAS-DInSAR ENVISAT data and Bayesian source inference," *Geophys. Res. Lett.*, vol. 35, no. 7, Apr. 2008, Art. ID. L07308.
- [61] J. Fernandez *et al.*, "Gravity-driven deformation of Tenerife measured by InSAR time series analysis," *Geophys. Res. Lett.*, vol. 36, no. 4, Feb. 2009, Art. ID. L04306.
- [62] F. Guzzetti *et al.*, "Analysis of ground deformation detected using the SBAS-DInSAR technique in Umbria, Central Italy," *Pure Appl. Geophys.*, vol. 166, no. 8/9, pp. 1425–1459, Sep. 2009.
- [63] J. Hunstad *et al.*, "Surface deformation in the Abruzzi region, Central Italy, from multi-temporal DInSAR analysis," *Geophys. J. Int.*, vol. 178, no. 3, pp. 1193–1197, Sep. 2009.
- [64] J. Ruch *et al.*, "Stress transfer in the Lazufre volcanic area, central Andes," *Geophys. Res. Lett.*, vol. 36, no. 6, Nov. 2009, Art. ID. L22303.
- [65] R. K. Ahuja, T. J. Magnanti, and J. B. Orlin, *Network flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [66] D. Bertsekas and P. Tseng, RELAX-IV: A faster version of the RELAX code for solving minimum cost flow problems. Dept. Elect. Eng. Comput. Sci., MIT, Cambridge, MA, USA, Tech. Rep., 1994.
- [67] B. Chapman, G. Jost, and R. Pas van der. *Using OpenMP: Portable Shared Memory Parallel Programming*. Cambridge, MA, USA: MIT Press, 2007.
- [68] S. Elefante *et al.*, "SBAS-DInSAR Time series generation on cloud computing platforms," in *Proc. IEEE IGARSS*, Melbourne, Australia, Jul. 2013, pp. 274–277.
- [69] A. Goscinski, R. Buyya, and J. Broberg Eds., *Cloud Computing: Principles and Paradigms*. Hoboken, NJ, USA: Wiley, 2011.
- [70] A. Pepe, B. Ortiz, P. R. Lundgren, P. A. Rosen, and R. Lanari: "The stripmap-ScanSAR Sbas approach to fill gaps in stripmap deformation time-series with ScanSAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4788–4804, Dec. 2011.
- [71] E. Sansosti, F. Casu, M. Manzo, and R. Lanari, "Space-borne radar interferometry techniques for the generation of deformation time series: An advanced tool for Earth's surface displacement analysis," *Geophys. Res. Lett.*, vol. 37, Nov. 2010, Art. ID. L20305.
- [72] E. Sansosti *et al.*, "How second generation SAR systems are impacting the analysis of ground deformation," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 28, no. 1, pp. 1–11, May 2014.
- [73] P. Tizzani *et al.* "Magma and fluid migration at Yellowstone Caldera in the last three decades inferred from InSAR, leveling and gravity measurements," *J. Geophys. Res., Solid Earth*, vol. 120, no. 4, pp. 2627–2647, Apr. 2015.



**Pasquale Imperatore** (M'10) received the Laurea degree (*cum laude*) in electronic engineering and the Ph.D. degree in electronic and telecommunication engineering from the University of Naples Federico II, Naples, Italy, in 2001 and 2011, respectively.

For four years, he was a Research Engineer with WISE S.p.A., Naples, where he worked on modeling and simulation of wave propagation, advanced ray-tracing-based prediction tool design for wireless applications in urban environments, and simulation and processing of synthetic aperture radar (SAR) signals. From 2005 to 2007, he was a Senior Researcher with the international research center CREATE-NET, Trento, Italy, where he conducted research and experimentation on radio-wave propagation at 3.5 GHz and on emerging broadband wireless technologies. Since 2007, he has been with the Department of Biomedical, Electronic and Telecommunication Engineering, University of Naples Federico II. Since 2011, he has been with the Institute for Electromagnetic Sensing of the Environment (IREA), Italian National Council of Research (CNR), Naples. His research interests include microwave remote sensing and electromagnetics, particularly scattering in random layered media, perturbation methods, parallel computing in electromagnetics, SAR data modeling and processing, SAR interferometry, radio-localization in urban environment, and electromagnetic propagation modeling, simulation, and channel measurement.

Dr. Imperatore serves as a Reviewer for several international journals.



**Antonio Pepe** (M'12) received the Laurea degree in electronic engineering and the Ph.D. degree in electronic and telecommunication engineering from the University of Naples Federico II, Naples, Italy, in 2000 and 2007, respectively.

After his graduation, following a short experience at Wind Telecommunication Spa, he joined in 2001 the Institute for Electromagnetic Sensing of the Environment (IREA), Italian National Research Council (CNR), Naples, where he currently holds a permanent position as a Researcher. In 2005, he was a Visiting Scientist with the University of Texas at Austin, Austin, TX, USA; in 2009, with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA; and in 2014, with East China Normal University, Shanghai, China. Since 2012, he has also been an Adjunct Professor of signal theory with the Università della Basilicata, Potenza, Italy. More recently, he has developed research activities for the generation of differential synthetic aperture radar interferometry (DInSAR) products through new-generation SAR instruments and for the generation of mixed ScanSAR-Stripmap DInSAR analyses. His main research interests include the development of advanced DInSAR algorithms aimed at monitoring surface deformation phenomena induced by subsidence, volcano activities, and earthquakes, with a particular interest toward the phase unwrapping problems.

Dr. Pepe serves as a Reviewer for several peer-reviewed international journals. He is a member of the Editorial Board of the *Advances in Geology* and the *Asian Journal of Geoscience*. He received the Best Reviewer Mention of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2014.



**Riccardo Lanari** (M'91–SM'01–F'13) received the Laurea degree in electronic engineering (*summa cum laude*) from the University of Naples, Federico II, Naples, Italy, in 1989.

In the same year, he joined the Research Institute for Electromagnetism and Electronic Components (IRECE) and then the Institute for Electromagnetic Sensing of the Environment (IREA), both Research Institutes of the Italian Council of Research (CNR), Naples. Since November 2011, he has been the Director of IREA-CNR. He has been a Visiting Scientist at different foreign research institutes, including the Institute of Space and Astronautical Science (ISAS), Japan (1993), the German Aerospace Research Establishment (DLR), Germany (1991 and 1994), and the Jet Propulsion Laboratory (JPL), CA, USA (1997, 2004, and 2008). From 2000 to 2003, he was an Adjunct Professor of electrical communications with the University of Sannio, Benevento, Italy. From 2000 to 2008, he was a Main Lecturer with the Institute of Geomatics, Barcelona, Spain. Moreover, he has recently achieved the national scientific habilitation as a Full Professor of telecommunications (in December 2013) and as a Full Professor of geophysics (February 2014). He is the author or coauthor of *Synthetic Aperture Radar Processing* (CRC Press, 1999) and more than 80 international journal papers which have, nowadays, 6000 citations (source: Google Scholar). He is a holder of two patents. His main research interests include microwave remote sensing field, particularly for what attains the development of SAR and InSAR digital data processing methods.

Dr. Lanari has lectured in several national and foreign universities and research centers and served as the Chair and/or scientific program committee member of many international conferences. He received from NASA a recognition (1999) and a group award (2001) for his activities related to the SRTM project based on the InSAR experiments carried out in 2000, within a mission of the Space Shuttle Endeavour. He is a Distinguished Speaker of the Geoscience and Remote Sensing Society.