A PARALLEL COMPUTATIONAL MODEL FOR MULTICHANNEL PHASE UNWRAPPING PROBLEM

Pasquale Imperatore, Antonio Pepe, Riccardo Lanari

IREA-CNR, Via Diocleziano 328, 80124 Napoli, Italy,
Email: imperatore.p@irea.cnr.it; pepe.a@irea.cnr.it; lanari.r@irea.cnr.it

ABSTRACT

In this paper, a parallel model for the solution of the computationally intensive multichannel phase unwrapping (MCh-PhU) problem is proposed. Firstly, the Extended Minimum Cost Flow (EMCF) algorithm for solving MCh-PhU problem is revised within the rigorous mathematical framework of the *discrete calculus*; thus permitting to capture its topological structure in terms of meaningful discrete differential operators. Secondly, emphasis is placed on those methodological and practical aspects, which lead to a parallel reformulation of the EMCF algorithm. Thus, a novel dual-level parallel computational model, in which the parallelism is hierarchically implemented at two different (i.e., process and thread) levels, is presented. The validity of our approach has been demonstrated through a series of experiments that have revealed a significant speedup. Therefore, the attained high-performance prototype is suitable for the solution of large-scale phase unwrapping problems in reasonable time frames, with a significant impact on the systematic exploitation of the existing, and rapidly growing, large archives of SAR data.

1. INTRODUCTION

The Multi-Channel Phase Unwrapping (MCh-PhU) problem constitutes the most critical task of the multichannel InSAR techniques from a computational point of view, due to its extremely time-consuming nature. Accordingly, MCh-PhU practical application to large data sets poses some challenges, yet, due to the associated intensive computational and storage requirements. Within this context, the Extended Minimum Cost Flow (EMCF) algorithm [1]-[2], which provides an effective strategy to solve the MCh-PhU problem, is considered. Firstly, EMCF is revised and casted within suitable matrix formalism, and then its computational structure is detailed to systematically gain insight into the computational structure of the problem. The rationale of a parallel reformulation for the EMCF algorithm is then presented, by 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. It is worth emphasizing that the proposed parallel scheme is oriented to take full advantage from the computational power offered by the emerging class of high performance computer (HPC) platforms [3]-[5], thus encompassing both distributed-memory and shared-memory parallelism. Several important aspects (dependence analysis, task-partitioning, scheduling strategy, etc.) related to our design strategy are discussed, also emphasizing the methodological approach and its implications. Practical implementation (in terms of software processes and threads) of the proposed parallel model is addressed, also clarifying the specific embraced models and tools. Performance evaluation relevant to the obtained prototype solution is carried out, thus quantifying the benefit of parallelism at different levels. A significant speed-up has been obtained with a consequent dramatic reduction of the relevant execution time, so experimentally demonstrating the validity of the presented approach. The experimental analyses have been also supported by analytical performance models, which have thus permitted us to shed light on pertinent foremost inefficiency factors. Furthermore, it has been proved that our solution permits us to maintain efficiency while increasing the number of involved computing elements (scalability goal). Therefore, the obtained prototype makes it possible the solution of the MCh-PhU problem [2] in a timely fashion, with an enormous impact on application exploitation, thus making practical the systematic investigation of available large-volume SAR data (including the ones collected by the new Sentinel-1 sensor), conveying information about complex Earth's crust deformation processes, for geospatial phenomena monitoring and understanding.

2. THE EMCF FORMULATION WITHIN THE DISCRETE CALCULUS FRAMEWORK

A graph-based description of the MCh-PhU problem naturally arises from its underlying discrete irregular data structure. We highlight that the gradient-based formulations of MCh-PhU approaches typically rely on a discrete setting based description [7]-[9]; 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. In fact, as far as discrete settings (e.g., graphs) are concerned, resorting to the conventional vectorial calculus might not be adequate. Accordingly, we here methodologically present the formulation of the EMCF algorithm within the purview discrete calculus; a comprehensive theoretical discussion is provided in [2].

In order to introduce the problem at hand, let us consider a set of Q single-look complex (SLC) SAR images acquired over an area of interest, with one of them selected as "reference master image" with respect to which all other images are co-registered. Two inherent vectors, representing the vectors of time acquisitions $\mathbf{t}=[t_1,\dots,t_Q]^T$ and of the perpendicular baselines $\mathbf{b}_\perp=[b_{\perp 1},\dots,b_{\perp Q}]^T$, respectively, characterize the SAR data set. Accordingly, taking into account the distribution of SAR acquisitions in the (temporal baseline, perpendicular baseline) Euclidean plane [1], a sequence of M multi-temporal SLC data pairs is first identified, forming in such a plane a Delaunay triangulation (i.e., the graph $\mathcal{G}_A=(\mathcal{V}_A;\mathcal{E}_A)$, where the set of vertexes $\mathcal{V}_A$ is related to SLC acquisitions and the set of edges $\mathcal{E}_A$ to SAR data pairs), and a corresponding sequence of multilook phase interferograms is subsequently generated. The MCh-PhU problem is inherently framed on a discrete setting: in fact, it is performed only on a discrete set of coherent pixels that are common to all the generated multilook interferograms. On these pixels a second graph, namely $\mathcal{G}_B=(\mathcal{V}_B;\mathcal{E}_B)$, is defined in the (azimuth, range) Euclidean plane. The topological properties inherent to each graph are algebraically captured by the related differential operators: the *discrete-gradient* operators, $\boldsymbol{\Pi}_A$ and $\boldsymbol{\Pi}_B$, and the *discrete-curl* operators, $\boldsymbol{\Omega}_A$ and $\boldsymbol{\Omega}_B$, respectively [2].

Firstly, a profitable linear phase model is introduced by using a valued functional description [1]-[2], including contributions due to residual topography Δz and surface displacement mean velocity Δv pertinent to the generic edge of the graph $\mathcal{G}_B$:

$$\boldsymbol{\Theta}(\Delta z,\Delta v)=\frac{4\pi}{\lambda}\left(\frac{\Delta z}{r\sin\vartheta}\boldsymbol{\Pi}_A\mathbf{b}_\perp+\Delta v\,\boldsymbol{\Pi}_A\mathbf{t}\right)\qquad(1)$$

where ϑ is the sensor look angle, λ is the operational wavelength and r is the sensor-to-target distance. The rationale of the model-based approach originally proposed in [1] is then to subtract the modelled contribution from the interferometric phase to reduce the relevant phase variations, so facilitating the subsequent PhU operations. Accordingly, $\forall n\in\{1,2,\dots,N\}$, we get:

$$\widetilde{\mathbf{g}}_n^T(\Delta z,\Delta v)=\boldsymbol{\Theta}(\Delta z,\Delta v)+W(\mathbf{g}_n^T-\boldsymbol{\Theta}(\Delta z,\Delta v))\qquad(2)$$

where $\widetilde{\mathbf{g}}_n^T$ is an estimate of the absolute discrete-gradient interferometric phase inherent to the generic n-th edge of the $\mathcal{G}_B$ graph (consisting of N edges) considered, and W represents the modulo-2π wrapping operator. The problem is then directly formulated as a set of N minimization procedures, with respect to the integer-valued vector $\mathbf{k}\in\mathbb{Z}^M$, whose elements represent the 2π integer multiples corrections that have to be applied in order to get the (corrected) unwrapped discrete-gradient. The problem to be solved assumes the form:

$$\hat{\mathbf{k}}_n^{(s)}=\underset{\substack{\mathbf{k}\in\mathbb{Z}^M\\(\Delta z,\Delta v)\in\Omega^{(s)}}}{\arg\min}\ \|\mathbf{k}\|_1\qquad(3a)$$

subject to:

$$\boldsymbol{\Omega}_A^T\mathbf{k}^T=\left\lfloor\boldsymbol{\Omega}_A^T\widetilde{\mathbf{g}}_n^T(\Delta z,\Delta v)(2\pi)^{-1}\right\rfloor\qquad(3b)$$

where $\Omega^{(s)}$ is a suitable bounded domain constraining the feasibility region for the searched Δz and Δv unknowns, and $\boldsymbol{\Omega}_A$ is the edge-face incidence matrix of the graph $\mathcal{G}_A$ (see [2] for more details).

The problem in (3a)-(3b) is a mixed-integer nonlinear programming problem (MILNP) for which efficient solutions are available. In particular, exploiting the underlying network structure of the problem, it can be formulated as a classical minimum cost flow (MCF) problem and casted efficiently using currently available solvers [6]. The estimated solution is then applied at first to correct the original matrix $\widetilde{G}=[\widetilde{\mathbf{g}}_1^T,\dots,\widetilde{\mathbf{g}}_N^T]^T$, so obtaining the estimated matrix $\hat{F}^{(s)}=\left[\hat{f}^{1(s)},\hat{f}^{2(s)},\dots,\hat{f}^{M(s)}\right]=\widetilde{G}-2\pi\hat{K}^{(s)}$, which represents the starting point for the second PhU operations, independently performed on the M interferograms over the graph $\mathcal{G}_B$. Accordingly, the second stage of the EMCF procedure is formulated as a set of M constrained minimization problems with integer variables, so that the current integer matrix describing the (unknown) residual normalized corrections, here denoted with $\mathbf{h}\in\mathbb{Z}^N$, are calculated as follows:
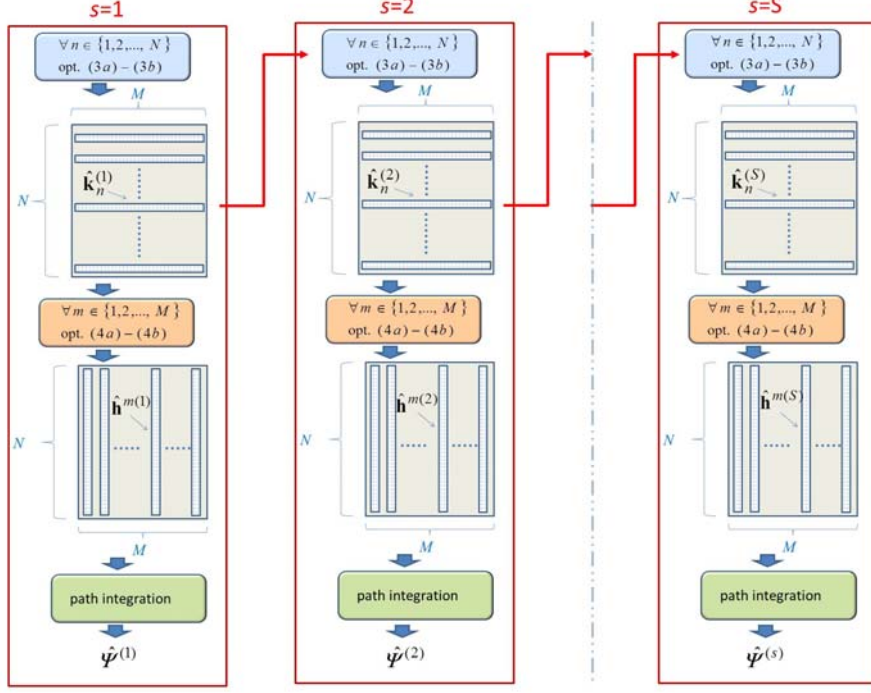
Fig. 1. Sequential version of EMCF Algorithm: Schematic Representation

$$\forall m \in \{1,2,...,M\},$$

$$\hat{\mathbf{h}}^{m(s)} = \underset{\mathbf{h} \in \mathbb{Z}^N}{\operatorname{argmin}} \left\| \mathbf{h} \right\|_{1,\mathbf{w}^{(s)}} \qquad (4a)$$

subject to:

$$\boldsymbol{\Omega}_{\mathrm{B}}^{\mathrm{T}} \mathbf{h} = \boldsymbol{\Omega}_{\mathrm{B}}^{\mathrm{T}} \hat{\mathbf{f}}^{m(s)} (2\pi)^{-1} \qquad (4b)$$

where $\mathbf{w}^{(s)} = [w_1^{(s)},\ldots,w_N^{(s)}]^{\mathrm{T}}$ is a proper vector that gives a confidence measure of the correctness of the phase estimation carried out in the first (temporal) stage of the phase unwrapping scheme [1]. Finally, taking into account the corrections resulting from the first and the second stage (3) and (4), we retrieve an estimate of the unwrapped phase gradients on the graph $\mathcal{G}_{\mathrm{B}}$:

$$\hat{\mathbf{S}}^{(s)} = \hat{\mathbf{F}}^{(s)} - 2\pi\hat{\mathbf{H}}^{(s)} = \tilde{\mathbf{G}} - 2\pi\hat{\mathbf{K}}^{(s)} - 2\pi\hat{\mathbf{H}}^{(s)} \qquad (5)$$

that are subsequently spatially integrated, for each single SAR data pair, to recover the unwrapped phase at the coherent targets' location (representing the vertexes of the graph $\mathcal{G}_{\mathrm{B}}$). The obtained solution, however, may be not the optimal one, due to the highly nonlinear character of the considered problem and the presence of many potential local minima. Accordingly, we adopted an optimization strategy based on a proper combination of several (independent) solutions obtained by solving the problems (3)-(4) on nested bounded subdomains

$\Omega^{(S)} \subset \ldots \subset \Omega^{(2)} \subset \Omega$, so allowing $(\Delta z, \Delta v)$ to vary in a prescribed searching 2-D space Ω. A more detailed description of the optimization procedure as well as of the adopted combination procedure of the independently retrieved unwrapped solutions is provided in [2]. The resulting computational structure of the sequential implementation of the EMCF algorithm is schematically depicted in Figure 1.

3. PARALLEL FORMULATION OF THE EMCF ALGORITHM

In this section the emphasis in on the parallel formulation of the EMCF algorithm previously discussed (Sect. 2). First of all, we emphasize that the proposed parallel scheme encompasses both distributed-memory and shared-memory parallelism, since it is oriented to take full advantage from the computational power offered by the emerging class of high performance computer (HPC) architectures. As far as the parallelization strategy is concerned, we have decomposed the problem at hand into single elements (sub-problems) that can be executed concurrently. Recasting the original algorithm into a form (decomposition pattern) [4]-[6] that exposes available concurrency inherent into the problem has been our primary concern. Flexibility, simplicity and efficiency/scalability have been three main guidelines adopted for the decomposition pattern strategy. The challenge was 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 problem at hand, which hinder direct concurrency exploitation, have been carefully identified and correctly managed.

TABLE I. COMPUTATIONAL PLATFORM CHARACTERISTICS

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

Then, the formulation of a *dual-level* parallel model has been obtained, by decomposing the computational problem into a hierarchical collection of components (tasks/subtasks) that can be executed concurrently, so expressing the nested parallelism that can be implemented at different levels. It is worth mentioning that EMCF computational structure includes intrinsic data-flow dependences between different macro-blocks (denoted by $s=1,...,S$), as it is evident by inspection of the algorithm pattern of Fig.1. Due to these ordering constraints between computationally significant parts of the EMCF algorithm (see red arrows in Fig 1), the problem at first glance appears to be inherently sequential. 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 detail in [2]. As far as efficiency (i.e. effectively use 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. Subsequently, the relevant prototype implementation has taken into account the distribution of the computational load among the computing units (processors/processor cores), so focusing on pertinent mapping and task scheduling for software implementation. Accordingly, practical implementation of the parallel scheme in the form of separate software *processes* and *threads* has been addressed: mapping parallelism on architectural resources, however, implies some insight into specific programming models and tools. Specifically, the associate parallel prototype envisages both messages passing, for communication between computing nodes, and multithreading (within a multiprocessor node) approaches to parallel programming. Furthermore, we emphasize that prototype implementation have also required additional complexity inherent to design and

programming of an efficient parallel algorithm for multiple computing units (processors/cores), in order to attain meaningful parallel performance in terms of speedup and execution time. Finally, we emphasize that parallel EMCF solution strategy implies the concurrent execution of different instances of the minimum cost flow (MCF) algorithm; however, the available RELAX-IV routine does not guarantee consistent concurrent-execution with thread-parallelism. Thus, we developed a thread-safe version of RELAX-IV. A more detailed discussion is provided in [2].

4. EXPERIMENTAL RESULTS

We selected as test-site the Naples Bay area, a volcanic and densely urbanized district in Southern Italy. We processed a SAR data set consisting of 64 acquisitions collected over ascending orbits by the ASAR/ENVISAT sensor. From these images, 175 small baseline interferograms, characterized by a (perpendicular) baseline value smaller than 400 m and a maximum temporal separation of about 1500 days, have been generated. With reference to the considered (fixed) problem, we conduct a performance analysis aimed at investigating the effectiveness of the proposed solution at the different levels of parallelism, in terms of speedup and scalability. The computational infrastructure we employ is described in Tab. I. As far as the analysis of the (distributed-memory) process-level parallelism is concerned, the speedup has been experimentally carried out on up to 16 processors and is shown (static scheduling) in Fig. 2 as a function of the number of used processors. As a matter of fact, high performances are achieved by exploiting the available distributed-memory parallelism, as evident from Fig. 2. The speedup predicted via a simple performance model is also depicted in Fig.2, thus gaining further insight in the main residual inefficiencies (in this case they are essentially ascribable to the residual load-unbalancing closely connected with the coarse granularity implied by the top-level of the parallel scheme). Investigation on the attained performance (at the node level) pertinent to the (shared-memory) thread-level parallelism is now in order. Different scheduling scenarios (static (SS), guided (GS) and dynamic (DS)) for load balancing at thread-level have been implemented [5], and relevant performances are evaluated: achieved speedup versus the number of engaged processor cores is depicted for three scheduling strategies in Fig.3. As a matter of fact, in our case, DS strategy outperforms the other strategies, thus permitting to minimize the overall execution time and also showing remarkable scalability. Furthermore, adopting an appropriate performance model allow us to gain further insight into the attained performance: in this case the residual inefficiency (DS strategy) turns out essentially associated with thread management overhead and memory bandwidth contention among multiple thread.
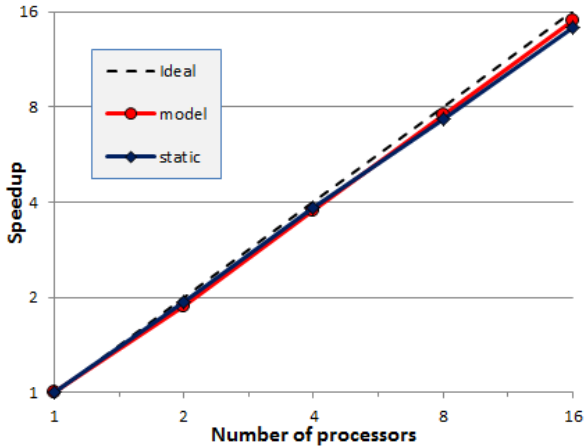
Figure 2. Multi-node performance: Speedup as a function of the number of engaged processors. The ideal linear speedup of and the speedup predicted by the considered performance model are also depicted.
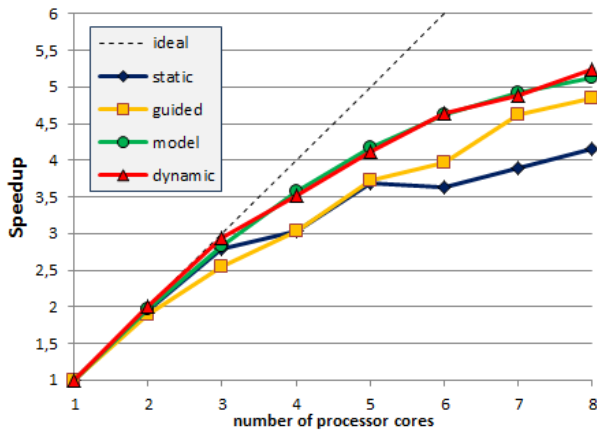

Figure 3. Multithreading performance: Speedup as a function of the number of engaged processor cores, obtained for different scheduling strategies. The ideal linear speedup and the speedup predicted by the considered performance model are also depicted.

5. CONCLUSIONS

We have proposed a novel parallel computational model for the specific EMCF solution strategy, to face the time-consuming nature of MCh-PhU problem. 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 [4]; this includes both shared-memory and distributed-memory architectures. Furthermore, we have experimentally demonstrated the validity of our approach. Performance evaluation relevant prototype

solution has been carried out, thus quantifying the benefit of parallelism at different levels. The obtained significant speedup implies a subsequent dramatic reduction of the relevant execution time; in addition, the proposed solution permits to maintaining efficiency while increasing number of engaged computing elements (scalability).

7. REFERENCES

1. A. Pepe, R. Lanari R., "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, 2006.

2. P. Imperatore, A. Pepe, R. Lanari, "Multichannel Phase Unwrapping: Problem Topology and Dual-Level Parallel Computational Model," *IEEE Trans. Geosci. Remote Sens.* (submitted).

3. F. Gebali, *Algorithms and Parallel Computing*, John Wiley & Sons, Inc., 2011.

4. H. El-Rewini and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*, John Wiley & Sons, Inc, 2005.

5. B. Chapman, G. Jost, and R. Pas van der, *Using OpenMP: Portable Shared Memory Parallel Programming*, MIT Press, 2007.

6. D. Bertsekas, P. Tseng, "The relax codes for linear minimum cost network flow problems," Annals of Operations Research, V. 13, 1988.

7. D. C. Ghiglia, M. D. Pritt, Two-dimensional phase unwrapping: theory, algorithms and software, New York, John Wiley, 1998.

8. M. Costantini, "A novel phase unwrapping method based on network programming," IEEE Trans. Geosci. Remote Sensing, vol. 36, pp. 813–821, May 1998.

9. R. Bamler, P.Hartl, "Synthetic aperture radar interferometry", Inverse problems, vol. 14, no.4, R1, 1998.