# PURE: Scalable Phase Unwrapping with Spatial Redundant Arcs

**Ravi Lanka**
Jet Propulsion Laboratory
California Institute of Technology
ravi.kira@jpl.nasa.gov

## Abstract

Phase unwrapping is a key problem in many coherent imaging systems, such as synthetic aperture radar (SAR) interferometry. A general formulation for redundant integration of finite differences for phase unwrapping (Costantini et al., 2010) was shown to produce a more reliable solution by exploiting redundant differential estimates. However, this technique requires a commercial linear programming solver for large-scale problems. For a linear cost function, we propose a method based on Dual Decomposition that breaks the given problem defined over a non-planar graph into tractable sub-problems over planar subgraphs. We also propose a decomposition technique that exploits the underlying graph structure for solving the sub-problems efficiently and guarantees asymptotic convergence to the globally optimal solution. The experimental results demonstrate that the proposed approach is comparable to the existing state-of-the-art methods in terms of the estimate with a better runtime and memory footprint.

## 1 Introduction

Phase unwrapping is the process of recovering unambiguous phase values from multi-dimensional phase data that are measured modulo $2\pi$ (wrapped data) and is affected by random noise and systematic disturbances. The difference between the measured and the actual phase has an ambiguity is integer multiples of $2\pi$. For unwrapping purposes it is usually assumed that the sampling rate is adequate over most of the data set so that aliasing is avoided. In other words, the true absolute phase difference between two neighboring data points is generally less than $\pi$. This reduces the phase unwrapping problem into that of integration of the phase difference between neighboring data points under certain paths (in noisy conditions). This problem is representative of a class of important imaging techniques such as interferometric synthetic aperture radar (Allen, 2008), optical interferometry (Pandit et al., 1994), magnetic resonance imaging (Lan et al., 2008), and diffraction tomography (Devaney, 1984). The advent of synthetic aperture radar interferometry (InSAR) (Rosen et al., 2000) and numerous other applications spurred interest in developing reliable two-dimensional (2D) phase unwrapping algorithms. The most widely used technique for 2D phase unwrapping relies on network programming approaches, which formulates the problem as a Minimum Cost Flow (MCF) (Costantini, 1998) and is applicable only for planar graphs.

Some of the factors that influence the quality of phase unwrapping for SAR interferometry include (1) signal interference due to atmospheric conditions; (2) temporal decorrelation due to the changes of the scattering characteristics at different times; and (3) geometric decorrelation due to different imaging geometries arising from the long distances between the repeated orbits of the same satellite. To address these factors, several techniques (Hooper and Zebker, 2007;Costantini et al., 2010; Shanker and Zebker, 2010) were developed to process multiple conventional interferograms collected at different times from the same area. A general formulation for redundant integration technique for multi-temporal phase unwrapping (Costantini et al., 2010) exploits redundant information obtained

(C) 2017 California Institute of Technology. Government Sponsorship Acknowledged.

from any pairs of points (typically close together but not necessarily nearest neighbors), making it possible to obtain a solution robust to outliers and noise both in 2D and multidimensional cases. The general formulation includes standard phase unwrapping and finite difference integration techniques as special cases.

Edgelist algorithm (Shanker and Zebker, 2010) utilizes the information from the temporal dimension by replacing the MCF formulation with its dual formulation, where the closed loops are replaced by reliable edges as the base construct. This variant formulation enables the inclusion of external geodesic measurements as constraints. A step-wise 3D algorithm (Hooper and Zebker, 2007) solves the multi-temporal unwrapping in multiple stages, first temporally and subsequently uses relaxed temporal solution to constrain the solution in spatial dimensions.

Similar to MCF formulation (Costantini, 1998), Edge list algorithm and Linear Programming (LP) based redundant arcs formulation (Costantini et al., 2010) exhibit total unimodularity which enables us to solve the integer programming problem efficiently. However, for large-scale interferograms, despite using commercial solvers, these formulations do not scale better than MCF formulation (section 3). Also, the publicly available LP solvers perform poorly when compared against commercial solvers in handling large datasets (Meindl and Templ, 2012). In the rest of the paper, we refer to the LP based redundant arcs formulation as LPRA.

Decomposition methods employ message-passing between sub-problems to iteratively improve the solution of large-scale optimization problems (Santos, 2009). However, their use is unexplored for phase unwrapping techniques to the best of our knowledge. Our contributions are: (1) We show how decomposition methods (Rush and Collins, 2012) that break the given problem into tractable subproblems, can be employed to solve the LPRA efficiently (2) We propose a decomposition that asymptotically converges to a globally optimal solution and shows a significant runtime improvements over LPRA. Further, the decomposed sub-problems only requires an MCF solver for which numerous efficient implementations are publicly available (Kiraly and Kovacs, 2012). In this paper, we focus on 2D interferograms with spatial redundant arcs.

The remainder of the paper is organized as follows: The background and related work is discussed in section (2). An introduction to decomposition method based on Lagrangian relaxation and the necessary condition for asysmptotic optimality is described in sections (2.4) and (2.6) respectively. The validation methods and results are described in section (3). Conclusion with future directions are discussed in section (4).

## 2 Background and Related work

The general phase unwrapping formulation is reviewed in section (2.1). A brief description of the conditions necessary for solving phase unwrapping as a MCF in section (2.2). LPRA construction as a totally uni-modular formulation is described in section (2.3).

### 2.1 Phase Unwrapping Formulation

Consider a set of points in 2D and let the phase measured at a point $i$ be $\psi_i$. Phase Unwrapping involves extraction of the unwrapped phase value, $\Phi_i$, from $\psi_i$, which are related by

$$\Phi_i = \Psi_i + 2\pi n_i \tag{1}$$

where $n_i$ represent the integer number of the cycles that must be added to $\Psi_i$ to obtain the corresponding $\Phi_i$. In addition, it is convenient to define a directed graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, whose nodes are the set of grid points $i$ and an edge $e_{(i,j)} \in \mathcal{E}$ that connects the grid points $i$ and $j$. We define a new variable $\delta_{ij}$ to represent the integer flow along the directed edge $e_{(i,j)}$ such that

$$n_i - n_j + \delta_{ij} - \delta_{ji} = \delta'_{ij}\ ,\ \forall e_{(i,j)} \in \mathcal{E} \tag{2}$$

$$\text{where } \delta'_{ij} = \left[\frac{\psi_i - \psi_j}{2\pi}\right]$$

$[.]$ represents the nearest integer function

The phase unwrapping can then be stated as,

$$\min_{\delta} \sum_{e_{(i,j)} \in \mathcal{E}} c_{ij} |\delta_{ij}|^m \tag{3}$$

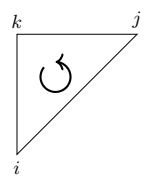
subject to,

$$\begin{aligned} n_i - n_j + \delta_{ij} - \delta_{ji} &= \delta'_{ij} \ , & \forall e_{(i,j)} \in \mathcal{E} \\ n_i, n_j &\in Z^+ & \forall \ i, j \in \mathcal{V} \\ \delta_{ij}, \delta_{ji} &\in \{0, 1\} & \forall e_{(i,j)} \in \mathcal{E} \end{aligned} \tag{4}$$

where the positive exponent *m* defines the selected metric, and $c_{ij} \in R$ representing the reliability of the preliminary estimates. When $m = 1$ ($m = 2$), the above problem reduces to a linear integer (quadratic) programming problem. The value of $n_i$ are determined only up to an additive constant because an additive constant leaves constraint (4) unaltered.

### 2.2 Phase Unwrapping as Network Programming

When the graph $G$ considered in equation (3) is planar, the constraint matrix is totally unimodular. In addition, the problem can be transformed to an equivalent MCF formulation in the dual network. Let $F$ be the faces in the planer embedding of $G$. Then, the constraints of the dual network is obtained by the set $F$ as follows. For each face $f \in F$, the edges constituting $f$ are traversed according to an uniformly chosen orientation and the constraints corresponding to the traversed edges are integrated together. As an example, consider the face shown below, comprised by the vertices $i, j, k$. Let the orientation be counter-clockwise.




The constraints of the edges $ij, jk, ki$ are

$$\begin{aligned} n_i - n_j + \delta_{ij} - \delta_{ji} &= \delta'_{ij} \\ n_j - n_k + \delta_{jk} - \delta_{kj} &= \delta'_{jk} \\ n_k - n_i + \delta_{ki} - \delta_{ik} &= \delta'_{ki} \end{aligned} \tag{5}$$

Traversing the edges in the order $ij, jk, ki$ and integrating the constraints, we get the reformulated constraint for MCF as

$$\begin{aligned} (\delta_{ij} - \delta_{ji}) + (\delta_{jk} - \delta_{kj}) + (\delta_{ki} - \delta_{ik}) \\ = \delta'_{ij} + \delta'_{jk} + \delta'_{ki} \end{aligned} \tag{6}$$

The integration path becomes

$$\begin{aligned} n_i - n_j + \delta_{ij} - \delta_{ji} &= \delta'_{ij} \\ n_j - n_k + \delta_{jk} - \delta_{kj} &= \delta'_{jk} \end{aligned} \tag{7}$$

Many different strategies (Kiraly and Kovacs, 2012) exist to solve MCF formulation. Supplementary constraints could be used to reflect a priori considerations, or could be useful in initializing the algorithm for improving the runtime.

### 2.3 Redundant arcs construction as LP

MCF formulation is applicable only to planar graphs. In the case of a general graph $\mathcal{G}$, Costantini et al., 2010 showed that it is sufficient to choose the cycle space of $\mathcal{G}$ as the basis and to sum the constraints of each cycle (in an uniformly chosen orientation) to transform the constraint matrix to a totally unimodular matrix. A numerically efficient way to construct this constraint space is by using a spanning tree. If the spanning tree is $\mathcal{S} := (\mathcal{V}, \mathcal{T})$ of $\mathcal{G}$, then each of the edges, $e_{(i,j)} \in \mathcal{E} \backslash \mathcal{T}$ are called backedges. Each such backedge forms a unique cycle in $\mathcal{G}$ called fundamental cycle.

This construction enables an optimal relaxation to the linear domain (for $m = 1$) through the use of totally unimodular property. Computationally efficient algorithms exist (Vanderbei, 2001) to solve redundant LP method, although slower than MCF. Our contribution in this work uses this construction to develop an iterative technique that is computationally more effective.

### 2.4 Our Approach

This section is organized as follows. A brief review of Dual Decomposition technique is presented in section (2.5). In section (2.6) and section (2.7), our method with the properties of the sub-problems for runtime improvement and asymptotic optimality are discussed respectively.

### 2.5 Dual Decomposition

We will first describe the general framework to illustrate methodology behind dual decomposition and then describe how this technique leads to an efficient way to solve phase unwrapping on a non-planar graph. Consider the following problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in Q \end{aligned} \tag{8}$$

where $Q$ is a closed convex set, $\mathbf{x}$ is a vector and $f$ is any function of $\mathbf{x}$. Suppose, the objective function $f$ is linearly separable (as in equation (3)), that is $f(\mathbf{x}) = \sum_i f_i(\mathbf{x}_i)$. Then, the above problem can be transformed using auxiliary public variable $z$ as,

$$\begin{aligned} \min_{\{\mathbf{x}_i\},z} \sum_i f_i(\mathbf{x}_i) \\ \text{subject to} \quad \mathbf{x}_i \in Q_i \quad \forall i \\ \mathbf{x}_i = P_i \mathbf{z} \quad \forall i \end{aligned} \tag{9}$$

where $Q_i$ is a closed convex set, such that $\cap_i Q_i = Q$. Vector $\mathbf{x}_i$ corresponding to each sub-problem (indexed by $i$) is the local copy of the public vector $\mathbf{z}$. $P_i$ is a binary matrix that maps the public vector $z$ into variables corresponding to each sub-problem $\mathbf{x}_i$. We can decouple the objective functions $f_i$ by relaxing the coupling constraint $x_i = P_i z$ using Lagrangian multipler ($\lambda$) to form the following dual function:

$$\begin{aligned} L(\mathbf{z}, \lambda) &= \min_{\{\mathbf{x}_i \in Q_i\},\mathbf{z}} \sum_i f_i(\mathbf{x}_i) + \lambda^T \Big(\mathbf{x} - P\mathbf{z}\Big) \\ &= \min_{\{\mathbf{x}_i \in Q_i\},\mathbf{z}} \sum_i \Big(f_i(\mathbf{x}_i) + \lambda_i^T \mathbf{x}_i\Big) - \lambda^T P\mathbf{z} \end{aligned} \tag{10}$$

Differentiating $L(z, \lambda)$ with respect to $z$ yields the condition $P^T\lambda = 0$. The dual function can now be solved independently for each $i$ given $\lambda$

$$\begin{aligned} L_i(\lambda_i) = \min_{x_i} f_i(\mathbf{x}_i) + \lambda_i^T \mathbf{x}_i \\ \text{subject to} \quad \mathbf{x}_i \in Q_i \end{aligned} \tag{11}$$

The dual of the equation (8) then becomes

$$\begin{aligned} \max_{\lambda} \quad & L(\lambda) = \sum_i L_i(\lambda_i) \\ \text{subject to} \quad & P^T\lambda = 0 \end{aligned} \tag{12}$$

The equation (12) is convex and can be solved with the projected subgradient method (Bertsekas, 2010). We refer to equation (12) as the dual decomposition master problem (DDMP). The projected subgradient algorithm is used to iteratively solve the DDMP algorithm, where at each iteration we first perform the decentralized optimization of the subproblems (in equation 11) followed by a sub-gradient update of the lagrangian multipliers ($\lambda$) as follows:

$$\lambda_i \leftarrow \lambda_i + \alpha_t g_{\lambda_i} \tag{13}$$

Here, $g_{\lambda_i}$ is the subgradient of the objective function $L_i$ with respect to $\lambda_i$ and $\alpha_t$ denotes the step size along the direction of positive subgradient. In our case, the subgradient of $g_\lambda$ is simply $x_i^*$, which is the optimal point $x_i$ in equation (11). The lagrangians $\lambda$ are then projected onto the constraint space in equation 12 given by:

$$\lambda_i \leftarrow \mathfrak{N}_\lambda(\lambda_i + \alpha_t x_i^*) \tag{14}$$

where, $\mathfrak{N}_\lambda$ denotes the projection function onto the set $\{\lambda | P^T\lambda = 0\}$. This process is repeated until convergence of the DDMP objective.

### 2.6 Definitions

In this section, we define the transformed phase unwrapping following the LPRA construction and the Lagrangian relaxation for the phase unwrapping problem more formally.

Without loss of generality, assume that the graph $\mathcal{G}$ is connected. Let $C$ be a sequence of arcs forming a closed path, and $\Gamma$ be a set of independent closed paths that span the whole cycle-space. Then, LPRA (Costantini et al., 2010) provides a numerically efficient way to solve the original phase unwrapping problem, by summing for each $C \in \Gamma$ the equation 4 that corresponds to arcs $(i, j)$ belonging to $C$ that transforms the constraints space to the following set of equations, denoted $\mathcal{A}(\Gamma)$:

$$\mathcal{A}(\Gamma): \qquad \sum_{(i,j)\in C} (\delta_{ij} - \delta_{ji}) = \sum_{(i,j)\in C} \delta'_{ij}, \qquad C \in \Gamma; \qquad \forall \delta_{ij}, \delta_{ji} \in \{0, 1\} \tag{15}$$

**Definition 1.** The transformed phase unwrapping on $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with cycle-space $\Gamma$, denoted $\mathbb{PU}$, is:

$$\mathbb{PU} := \min_{\substack{\delta\in\mathcal{A}(\Gamma)\\ \delta\in\{0,1\}}} \sum_{ij\in\mathcal{E}} c_{ij}\delta_{ij}$$

The linear minimization problem, denoted $\mathbb{LP}$, is obtained by linear relaxation of the integer flows:

$$\mathbb{LP} := \min_{\substack{\delta\in\mathcal{A}(\Gamma)\\ \delta\in[0,1]}} \sum_{ij\in\mathcal{E}} c_{ij}\delta_{ij}$$

**Definition 2.** With a little abuse of exponent notation, let $\mathcal{H} = \{\mathcal{G}^k(\mathcal{V}^k, \mathcal{E}^k) : k \in [1, \mathcal{K}]\}$ be a decomposition of a graph $\mathcal{G}$, where $\mathcal{K}$ is the cardinality of the set $\mathcal{H}$ that covers the graph $\mathcal{G}$ such that $\cup\mathcal{E}^k = \mathcal{E}$ and $\mathcal{V}^k = \mathcal{V}$. Let $\mathcal{G}_e(i, j)$ be the set of all subgraphs that contain edge $(i, j)$. Let $\mathcal{A}(\Gamma^k)$ be the constraint space defined on each subgraph $\mathcal{G}^k$, with cycle space $\Gamma^k \subset \Gamma$ (as in equation 15).

Assume that we introduce local flow variables and the corresponding cost function $\{\delta_{ij}^k, c_{ij}^k : ij \in \mathcal{E}_k\}$ for each subproblem $k$, with additional consistency constraints $\sum_k c_{ij}^k = c_{ij}$ and $\delta_{ij}^k = \delta_{ij}$. We can then define the Lagrangian relaxation for the $\mathbb{LP}$ by introducing a vector of lagrangian variable $\lambda$ for every consistency constraint on $\delta_{ij}^k$, denoted $\mathbb{LR}$, as:

$$\mathbb{LR} := \max_{\lambda} \sum_k f^k(\lambda^k)$$

$$f^k(\lambda^k) = \min_{\substack{\delta^k\in\mathcal{A}(\Gamma^k)\\ \delta^k,\delta\in[0,1]}} \sum_{ij\in\mathcal{E}^k} c_{ij}^k\delta_{ij}^k + \lambda_{ij}^k\left(\delta_{ij}^k - \delta_{ij}\right)$$

We can now state the theorem 1 behind LPRA algorithm's construction, which appears as Theorem 3.4c in (Kavitha et al., 2009).

**Theorem 1. (Tight Relaxation):** *Constraint matrix $\mathcal{A}$ defined on the cycle space $\Gamma$ is totally unimodular. Thus, if either $\mathbb{PU}$ or $\mathbb{LP}$ has a finite optimal value, then so does the other, and their optimal values coincide.*

Theorem 1 enables us to solve the original phase unwrapping on a non-planar graph as a linear programming problem. In other words, this theorem states that the linear relaxation $\mathbb{LP}$ is *tight*. We now develop an iterative algorithm for phase unwrapping.

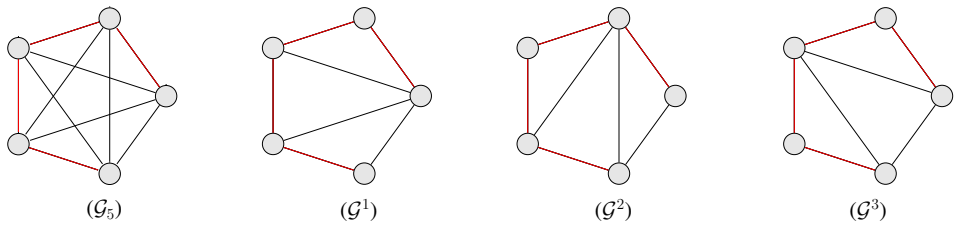



Figure 1: $\mathcal{G}_5$ is an example of phase unwrapping on a non-planar graph. The nodes and the edges of the graph correspond to the measured wrapped phase and the constraints respectively. The sub-graphs $\mathcal{G}^1$, $\mathcal{G}^2$ and $\mathcal{G}^3$ show an example decomposition of the non-planar graph. Red colored edges in $\mathcal{G}_5$ and each of subgraph $\mathcal{G}^1$, $\mathcal{G}^2$ and $\mathcal{G}^3$ corresponds to the chosen spanning tree.

### 2.7 Phase Unwrapping via Dual Decomposition

In our approach, we exploit the *strong* duality properties of any linear program to develop an iterative algorithm for $\mathbb{PU}$. Assume that the constraints of any linear program can be divided into "easy" and "hard" constraints. More specifically in the context of phase unwrapping, the constraints that break the planarity property of the underlying graph is classified as "hard" constraint. We can then use $\mathbb{LR}$ to transfer some of the constraints into the objective function. Thus, the easy constraints define $A(\Gamma^k)$. This leads us to the following theorem:

**Theorem 2. (Cycle Decomposition):** *For any graph decomposition that covers $\mathcal{G}$, $\mathbb{LR} \leq \mathbb{PU}$. Furthur, the equality strictly holds when $\cap_k \mathcal{A}(\Gamma^k) = \mathcal{A}(\Gamma)$.*

We refer the reader to the Appendix A for a proof. Theorem 2 allows us to break the $\mathbb{LP}$ thereby enabling us to use the underlying planarity structure and still retain asympthotic optimality. The asymptotic nature of the algorithm stems from the fact that the lagrangian variables $\lambda$ in the $\mathbb{LR}$ problem is iteratively updated through sub-gradient ascent (Rush and Collins, 2012).

In other words, to guarantee asymptotic optimality, we have to ensure that for each vertex pair belonging to an edge, there exists a path along a chosen spanning tree in atleast one of the subgraphs. We show an example decomposition of a non-planar subgraph $K_5$ in figure 1 in which the chosen spanning tree is shown in red. This ensures that the decomposition works on the $\mathbb{LP}$ formulation of the original problem (Costantini et al., 2010), thus inheriting asymptotic optimality. In addition, the planarity of each subgraph enables us to solve them efficiently as an MCF.

**Algorithm 1:** Phase Unwrapping with Redundant Arcs via Dual Decomposition

**Initialize**: dual variables $\lambda$ that satisfies $\sum_{k \in G_e(i,j)} \lambda_{ij}^k = 0 \; \forall \; ij \in \mathcal{E}$ (e.g., $\lambda = 0$)
**while** *Stopping criteria not met* **do**
  ▷ `Optimize each subproblem separately using MCF`
  **for** $k = 1$ *to* $\mathcal{K}$ **do**
    $\delta^{k\star} = \underset{\delta^k \in \mathcal{A}(\Gamma^k)}{\text{argmin}} \sum_k (c^k + \lambda^k)^T \delta^k$
  **end**

  ▷ `Update the dual variables`
  **for** $k = 1$ *to* $\mathcal{K}$ **do**
    **for** $(i,j) \in \mathcal{G}_e(i,j)$ **do**
      $\lambda_{ij}^k += \alpha_t \left( \delta_{ij}^{k\star} - \frac{1}{|\mathcal{G}_e(i,j)|} \sum_{q \in \mathcal{G}_e(i,j)} \delta_{ij}^{q*} \right)$
    **end**
  **end**
**end**

The constraint on the cost function (in definition 2) can be satisfied by distributing the cost ($c_{ij}$) uniformly among subgraphs. The algorithm (1) describes the pseudo-code for Dual Decomposition based phase unwrapping. It can be easily shown that the algorithm (1) corresponds to a projected subgradient ascent algorithm for solving $\mathbb{LR}$ problem.

The algorithm (1) consists of two main steps:

- Line 4: In the first step, the lagrangian variables are fixed to optimize each subproblem ($f^k(\lambda^k)$ in definition 2) as a MCF independently.
- Line 8-9: In the second step, the dual variables are updated using subgradient ascent, such that the consistency constraint (as in definition 2) for local flow variables belonging to each subproblem is satisfied.

These steps are repeated until convergence. We use the relative change in the dual objective value as the convergence criteria for the algorithm (1), since it is expected to plateau near the dual optimum.

## 3 Results and Discussions

**Methods Compared:** We first evaluate the performance of our method (henceforth referred to as PURE) using two different MCF algorithms, namely cost-scaling (Goldberg and Kennedy, 1995) and the network simplex algorithm (Orlin, 1997). The cost-scaling algorithm is a primal-dual method that applies a successive approximation scheme by scaling the cost. The network simplex algorithm is a specialized version of the simplex algorithm for linear programming that exploits the network structure of the MCF problem and performs the basic operations directly on the graph representation. RelaxIV (Bertsekas, 1994), a dual ascent based algorithm is another notable solving technique for MCF. However RelaxIV is not directly applicable to PURE, since it assumes an integral cost.

Our primary empirical results is the comparison of PURE against LPRA with variable redundant arcs. In order for the evaluation to be implementation independent, we report the running time for each algorithm as the time taken by the respective solvers. For completeness, we also present the quality improvements we obtain by using the redundant arcs compared to LPRA (with redundant arcs distance = 2).

**Experimental Setup:** We used a commercially solver Gurobi (Version 6.5.1) to solve the LPRA formulation. For solving MCF, we used the Cost-scaling code of Goldberg and Cherkassky, an efficient authoritative implementation of the algorithm. In addition, we used the Network scaling algorithm available in the MCFClass project (cite), a common C++ interface for several MCF solvers implementation.

The cost functions (in definition 2) is used to direct the placement of phase cycle jumps using quality measures derived from the data itself. The cost function is a crucial parameter that directly influences the quality of the solution. For our experiments, we used the smooth cost function as defined in SNAPHU (Chen and Zebker, 2001). In all our experiments on PURE, we scaled the cost in the objective function between [0, 1] and used a constant learning rate ($\alpha_t$ in 1) until change in objective value is below a threshold of $0.02$ and a decaying learning rate ($\alpha \leftarrow \frac{\alpha}{2}$) from there on with a convergence threshold of $0.001$.

**Datasets:** The experiment is conducted on a set of four SAR interferograms. For each of the interferogram, we generated $40$ different instance, $10$ instances each for every noise level from $0.4$ to $1$ with an increment of $0.2$. This allows us to study the running time performance of the algorithm with varying noise levels. We conduct the same set of experiments on two different resolutions $256 \times 256$ & $768 \times 768$ (henceforth referred to as setX, setY respectively) to study the effect of constraint set size on the running time. The number of constraints which depend on redundant arcs ranges from $100k$ to over $500k$ for setX and from $1$ million to over $5$ million for setY, which can be quite challenging to solve even for a commercial solver as we show later.

**Main Results:** In the first set of experiments, we conduct an empirical evaluation of the running time for PURE using cost-scaling and simplex algorithms. Figure 3a depicts the running time comparison as we vary the noise level for both redundant arcs {1,2} on setX. This plot is representative of the typical performance obtained in the setY instances. We observed that the performance of cost-scaling algorithm was more time efficient when compared to simplex algorithm for all our instances on setX. We are not aware of a theoretical interpretation of the performance difference at the time of writing

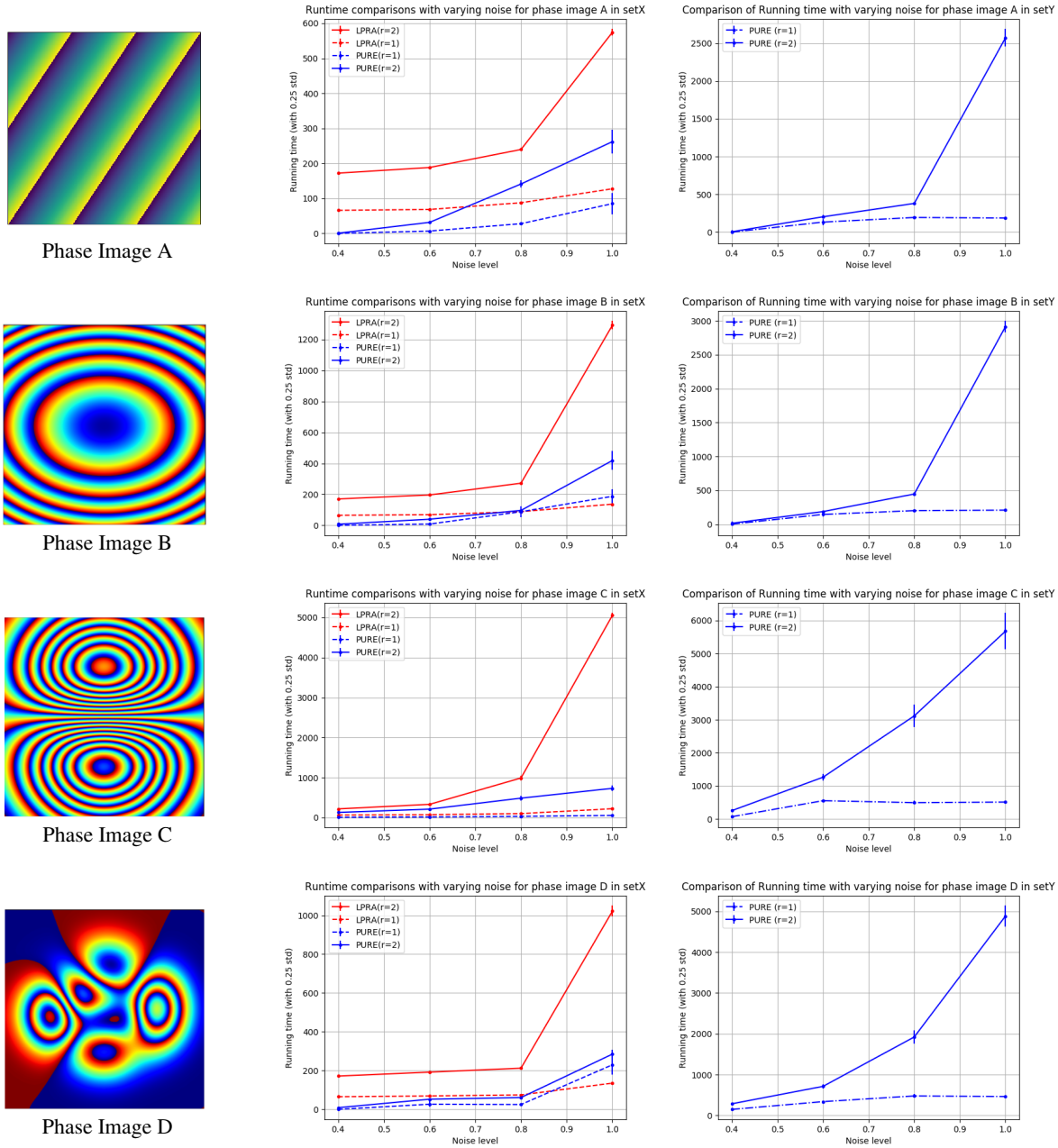
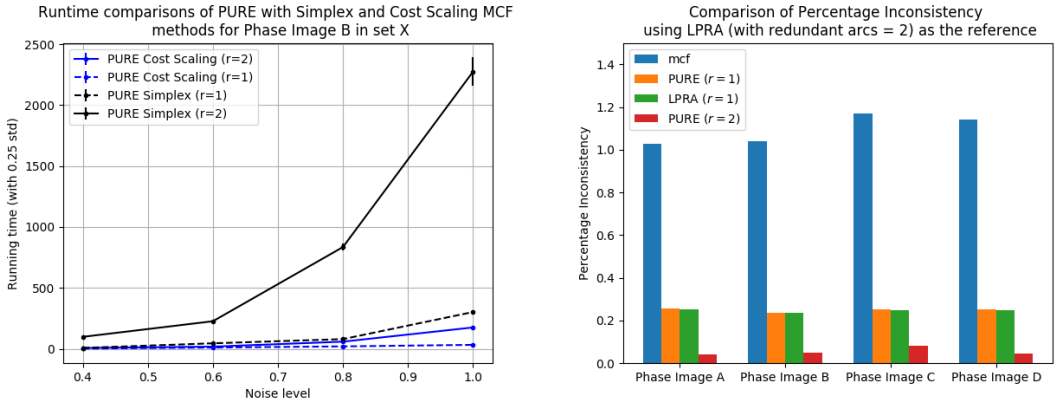



Figure 2: The first column above shows the wrapped phase images (A,B,C & D) without noise used in all our experiments. The plots in the second and third column show the running time (in seconds) comparison between PURE and LPRA for variable noise levels in the input. The value r corresponds to the redundant arcs $\{1, 2\}$. In all of the plots above, the dotted and thick line correspond to running time with redundant arcs 1 & 2 respectively.

the paper. For all our future experiments, we use PURE with the cost-scaling algorithm for solving the MCF subproblems.

In the second set of experiments, we validate the scalability of PURE by comparing the running time with LPRA. For this experiment, we predefined the planar decomposition for each redundant arcs level $\{1, 2\}$ that satisfies the condition for asymptotic optimality as in Theorem 2. First, we note that although all the problem instances in setX were solved by both the algorithms, PURE was several




(a) (b)

Figure 3: Figure (a) shows the Running time (in seconds) comparison of PURE with cost-scaling and simplex MCF algorithm. Figure (b) shows comparison of percentage inconsistency with unit noise variance using LPRA (with redundant arcs 2) as reference on setX. Value r in the legend corresponds to the redundant arcs length.

magnitudes faster in running time compared to LPRA. Secondly, LPRA failed to solve any of the instances in setY due to the large constraint set. To the best of our knowledge, PURE is the first algorithm to have solved constraint set of this scale for phase unwrapping with redundant arcs.

In order to further quantitatively evaluate the performances of different phase unwrapping methods, we also present the comparison of inconsistency measure in Figure 3b with LPRA with redundant arcs distance = 2 as reference. The inconsistency measure is defined as the percentage number of reconstructed phase cycle that matches with ground truth phase cycle. As expected, LPRA and PURE show significant improvement over the MCF based approach. In addition, PURE is comparable in performance to that of LPRA with significantly improved running time performance.

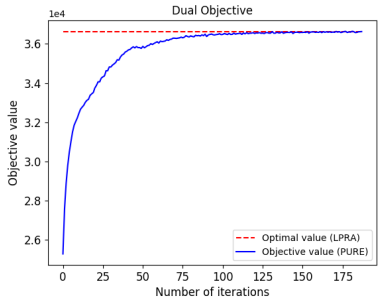



Figure 4: Evolution of dual objective value for an instance of wrapped phase B in setX.

## 4 Conclusion and Future Work

We proposed a scalable algorithm for phase unwrapping with redundant arcs and provide empirical evaluation of our performance on simulate SAR interferograms. In future, we aim to extend this work to the temporal dimension of the interferograms. The rate of convergence and its analysis (Rush and Collins, 2012) depend on the update schedule used by the algorithm. We would also like to analyze the rate of convergence under different variants of subgradient ascent techniques.

### Acknowledgments

We would like to thank Piyush Agram for introducing the authors to Phase Unwrapping and Redundant Arcs technique and for the numerous discussion that shaped the problem definition. In addition, the authors would also like to thank BalaSanjeevi and Arun Viswanathan for reviewing the manuscript. This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. All rights reserved.

## References

C T Allen. Interferometric Synthetic Aperture Radar. *IEEE Geoscience and Remote Sensing Society Newsletter*, (96):6–13, 2008.

Dimitri P Bertsekas. RELAX-IV : A Faster Version of the RELAX Code for Solving Minimum Cost Flow Problems by. pages 1–18, 1994.

Dimitri P Bertsekas. Incremental Gradient , Subgradient , and Proximal Methods for Convex Optimization : A Survey. 2010:1–38, 2010.

Curtis W Chen and Howard A Zebker. Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization. 18(2):338–351, 2001.

M Costantini. A novel phase unwrapping method based on network programming. *Proceedings of the IEEE Transactions on Geoscience and Remote Sensing*, 36(3):813–821, 1998.

Mario Costantini, Fabio Malvarosa, and Federico Minati. A general formulation for robust and efficient integration of finite differences and phase unwrapping on sparse multidimensional domains. *Proceedings of European Space Agency Fringe 2009 Workshop*, 2009(3):758–768, 2010.

A. J. Devaney. Geophysical diffraction tomography. *IEEE Transactions on Geoscience and Remote Sensing*, GE-22(1):3–13, Jan 1984. ISSN 0196-2892. doi: 10.1109/TGRS.1984.350573.

Andrew V. Goldberg and Robert Kennedy. An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 71(2):153–177, 1995. ISSN 14364646.

Andrew Hooper and Howard A. Zebker. Phase unwrapping in three dimensions with application to InSAR time series. *Journal of the Optical Society of America*, 24(9):2737–2747, 2007. doi: 10.1364/JOSAA.24.002737.

Telikepalli Kavitha, Christian Liebchen, Kurt Mehlhorn, Dimitrios Michail, Romeo Rizzi, Torsten Ueckerdt, and Katharina A. Zweig. Cycle bases in graphs characterization, algorithms, complexity, and applications. *Computer Science Review*, 3(4):199–243, 2009. ISSN 15740137. doi: 10.1016/j.cosrev.2009.08.001.

Z Kiraly and P Kovacs. Efficient implementations of minimum-cost flow algorithms. *Acta Universitatis Sapientiae, Informatica 4*, 1:67–118, 2012.

T. Lan, D. Erdogmus, S. J. Hayflick, and J. U. Szumowski. Phase unwrapping and background correction in mri. In *2008 IEEE Workshop on Machine Learning for Signal Processing*, pages 239–243, Oct 2008. doi: 10.1109/MLSP.2008.4685486.

B Meindl and M Templ. Analysis of commercial and free and open source solvers for linear optimization problems. 2012.

James B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129, 1997. ISSN 00255610. doi: 10.1007/BF02614365.

S. M. Pandit, N. Jordache, and G. A. Joshi. Data-dependent systems methodology for noise-insensitive phase unwrapping in laser interferometric surface characterization. *J. Opt. Soc. Am. A*, 11(10): 2584–2592, Oct 1994. doi: 10.1364/JOSAA.11.002584.

P. A. Rosen, S. Hensley, I. R. Joughin, F. K. Li, S. N. Madsen, E. Rodriguez, and R. M. Goldstein. Synthetic aperture radar interferometry. *Proceedings of the IEEE*, 88(3):333–382, March 2000. ISSN 0018-9219. doi: 10.1109/5.838084.

Alexander M Rush and Michael Collins. A Tutorial on Dual Decomposition and Lagrangian Relaxation for Inference in Natural Language Processing. 45:305–362, 2012.

Dos Santos. Finding Approximate Solutions for Large Scale Linear Programs. 2009.

A. Piyush Shanker and Howard A. Zebker. Edgelist phase unwrapping algorithm for time series InSAR analysis. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 27(3):605–612, 2010.

Robert J. Vanderbei. *Linear programming : foundations and extensions*. International series in operations research & management science. Kluwer Academic, 2001.

# Appendix

## A Proofs for Theoretical Results

**Theorem. (Cycle Decomposition):** *For any graph decomposition that covers $\mathcal{G}$, $\mathbb{LR} \leq \mathbb{PU}$. Furthur, the equality strictly holds when $\cap_k \mathcal{A}(\Gamma^k) = \mathcal{A}(\Gamma)$.*

*Proof.* Let us define $\overline{\mathbb{LP}}$ to facilitate the proof. From definition 2, $\Gamma^k \subset \Gamma$ which implies $\mathcal{A}(\Gamma) \subseteq \cap_k \mathcal{A}(\Gamma^k)$. Let $B(\Gamma)$ denote the additional set of constraints in $\mathcal{A}(\Gamma)$ such that, $\mathcal{A}(\Gamma) = \cap_k \mathcal{A}(\Gamma^k) \cap B(\Gamma)$. Then $\mathbb{LP}$ can be transformed to the following form:

$$\mathbb{LP} := \min_{\substack{\forall \text{k}, \delta^k \in \mathcal{A}(\Gamma^k);\ \delta^k = \delta \\ \delta \in \mathcal{B}(\Gamma);\ \delta, \delta^k \in [0,1]}} \sum_k c^{kT} \delta^k$$

$$\text{where,} \sum_{k \in G_e(i,j)} c^k_{ij} = c_{ij} \qquad \forall ij \in \mathcal{E}$$

We can then define $\overline{\mathbb{LP}}$ as follows:

$$\overline{\mathbb{LP}} := \min_{\substack{\forall \text{k}, \delta^k \in \mathcal{A}(\Gamma^k);\ \delta^k = \delta \\ \delta, \delta^k \in [0,1]}} \sum_k c^{kT} \delta^k$$

We note that $\overline{\mathbb{LP}} \leq \mathbb{LP}$ follows from the fact that the constraint space of $\mathbb{LP}$ is contained in the constraint space of that of $\overline{\mathbb{LP}}$. Equality constraint strictly holds when $\mathcal{B}(\Gamma) = R^n$, which implies $\mathcal{A}(\Gamma) = \cap_k \mathcal{A}(\Gamma^k)$. For ease of notation, we use $A^k \delta^k - b^k \leq 0$ to denote $\{\delta^k : \delta^k \in \mathcal{A}(\Gamma^k), \delta^k \in [0,1]\}$. Let us know consider the dual of $\overline{\mathbb{LP}}$, denoted dual-$\overline{\mathbb{LP}}$:

$$\text{dual-}\overline{\mathbb{LP}} := \max_{\mu, \lambda} \min_{\delta^k, \delta} \sum_k c^{kT} \delta^k + \mu^{kT}(A^k \delta^k - b^k) + \lambda^{kT}(\delta^k - \delta)$$

Upon further simplifying, we have

$$\text{dual-}\overline{\mathbb{LP}} := \max_{\mu \leq 0, \lambda} \sum_k -\mu^{kT} b^k$$

$$\text{subject to:} \qquad c^k + A^{kT} \mu^k + \lambda^k = 0 \quad \forall\, k$$

$$\sum_{k \in G_e(i,j)} \lambda^k_{ij} = 0 \quad \forall\, ij \in \mathcal{E}$$

Using linear programs strong duality property, $\overline{\mathbb{LP}} =$ dual-$\overline{\mathbb{LP}}$. Let $(\mu^*, \lambda^*)$ be the optimal solution of dual-$\overline{\mathbb{LP}}$. Then evaluating $\mathbb{LR}$ at $\lambda^*$, denoted $\mathbb{LR}_{\lambda^*}$, we have

$$\begin{aligned} \mathbb{LR}_{\lambda^*} &= \sum_k \min_{A^k \delta^k \leq d^k} [c^{kT} \delta^k + \lambda^{*kT} \delta^k] \\ &= \sum_k \min_{A^k \delta^k \leq d^k} -\mu^{*kT} A^k \delta^k \geq \sum_k -\mu^{*kT} b^k \end{aligned}$$

Thus, $\mathbb{LR}_{\lambda^*} \geq$ dual-$\mathbb{LP} = \mathbb{LP}$. We can also easily show that $\mathbb{LR} \leq \mathbb{LP}$. Thus $\mathbb{LR} = \mathbb{LP}$, when $\mathcal{A}(\Gamma) = \cap_k \mathcal{A}(\Gamma^k)$ and it follows from theorem 1, $\mathbb{LR} = \mathbb{PU}$.

A variant of this proof was presented in Santos [2009], however to the best of our knowledege this is the first extension to TUM.

□