A COMPUTATIONALLY EFFICIENT ALGORITHM FOR DISTRIBUTED ADAPTIVE SIGNAL FUSION BASED ON FRACTIONAL PROGRAMS

Cem Ates Musluoglu and Alexander Bertrand

KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Kasteelpark Arenberg 10, box 2446, 3001 Leuven, Belgium
{cemates.musluoglu, alexander.bertrand}@esat.kuleuven.be

ABSTRACT

Spatial filtering procedures aim to optimally fuse the different signals collected in a sensor array, by exploiting their inter-channel correlations. If the sensors are physically distributed, as it is the case in a wireless sensor network, the inter-channel statistics cannot directly be measured or tracked, unless the data is transmitted to a central processor, which is not always possible due to energy or bandwidth constraints. The so-called distributed adaptive signal fusion (DASF) algorithm allows to solve such problems in a distributed fashion with a reduced communication burden. The DASF algorithm iterates over the different nodes of the network, each solving a local compressed version of the original (centralized) optimization problem. However, if the solver for these local optimization problems is in itself also iterative, the computational burden can become quite large as these iterations are nested within the DASF iterations. In this paper, we focus on Dinkelbach's iterative procedure to solve fractional programs, i.e., problems of which the objective function is a ratio of two continuous functions. We propose the fractional DASF (F-DASF) algorithm which interleaves the iterations of DASF with those of Dinkelbach's procedure, to reduce the computational burden without affecting the convergence properties of the original DASF algorithm.

Index Terms— Distributed Optimization, Distributed Spatial Filtering, Fractional Programming.

1. INTRODUCTION

Spatial filtering consists of linearly combining signals measured at different locations such that the resulting filtered signal is optimal in some sense [1,2]. This technique is widely used in biomedical signal processing [3–5], wireless communication [6, 7], and acoustics [8, 9] among others. With the emergence of wireless sensor networks [10, 11], many applications require a fully distributed approach to solve spatial filtering problems in order to reduce the energy and bandwidth requirements.

The filter design is typically based on an optimization problem aiming at finding a spatial filter which optimally fuses the sensor channels of the different nodes such that the resulting fused output signal is optimal in some sense. We refer to this class of problems as distributed signal fusion optimization (DSFO) problems. Classical distributed signal processing algorithms [12–15] typically assume a per-node separable objective function, which is not the case for DSFO problems. Indeed, in the DSFO setting, the cost function typically requires the inter-channel second-order statistics between all the sensor channel pairs in the network. The latter cannot be measured or tracked directly in such a distributed setting, unless all the sensor data is transmitted to a fusion center.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 802895). The authors also acknowledge the financial support of the FWO (Research Foundation Flanders) for project G081722N, and the Flemish Government (AI Research Program).

Table 1: Examples of DSFO problems with fractional objectives as in (2). TRO is the trace ratio optimization problem and RTLS the regularized total least squares. $\mathbb{E}$ denotes the expectation operator.

Problem	Cost function to minimize	Constraints
TRO [18]	$-\frac{\mathbb{E}[\|\|X^T\mathbf{y}(t)\|\|^2]}{\mathbb{E}[\|\|X^T\mathbf{v}(t)\|\|^2]}$	$X^TX=I_Q$ $\Leftrightarrow (X^TB)(X^TB)^T=I_Q$ with $B=I_M$
RTLS [19, 20]	$\frac{\mathbb{E}[\|\|\mathbf{x}^T\mathbf{y}(t)-d(t)\|\|^2]}{1+\|\|\mathbf{x}\|\|^2}$	$\|\|\mathbf{x}^TL\|\|^2\leq\delta^2$

The distributed adaptive signal fusion (DASF) framework proposed in [16] allows to solve DSFO problems in a distributed way, achieving convergence to the centralized solution under mild constraints [17]. At each iteration, the DASF framework requires a node of the network to solve a local optimization problem, which inherits the structure of the original (centralized) problem, and which can therefore be solved using the same optimization algorithm.

In this work, we are interested in solving a specific class of DSFO problems, namely fractional problems, for which the objective function is written as a ratio of two functions, such as, e.g., the trace ratio optimization problem [18] or the regularized total least squares problem [19, 20]. These problems are commonly solved using the generic Dinkelbach procedure [21]. Since the Dinkelbach procedure is itself iterative, the DASF framework will be computationally expensive for fractional problems because of the presence of nested iterations to solve per-node fractional problems within the iterations of the DASF algorithm. To avoid this computational burden, we propose the fractional DASF (F-DASF) algorithm, which interleaves the steps of the Dinkelbach procedure with the ones of the DASF algorithm. Even though none of the nodes fully solves its local problem (i.e., they only perform a single iteration of Dinkelbach's procedure), the resulting F-DASF algorithm has a guaranteed convergence under the same assumptions as the original DASF algorithm. We also empirically show by means of simulations that the convergence rate is not affected.

2. PROBLEM SETTING

Let us consider a sensor network with K nodes with the node set denoted as $\mathcal{K}=\{1,\ldots,K\}$. The topology of the network is given by a graph $\mathcal{G}$. Each node measures its own M_k-channel signal $\mathbf{y}_k$ and the network-wide signal $\mathbf{y}$, assumed to be ergodic and (short-term) stationary, is defined as

$$\mathbf{y} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_K^T]^T. \quad (1)$$

We denote as $\mathbf{y}(t) \in \mathbb{R}^M$ the observation of $\mathbf{y}$ collected at sample time t, where $M = \sum_k M_k$. We aim to find a spatial filter $X \in \mathbb{R}^{M\times Q}$ which is the solution of a fractional problem with the generic form[1] (see Table 1 for some illustrative examples)

$$\begin{aligned}
\underset{X\in\mathbb{R}^{M\times Q}}{\text{minimize}} \quad & \varrho\left(X^T\mathbf{y}(t), X^TB\right) \triangleq \frac{\varphi_1\left(X^T\mathbf{y}(t), X^TB\right)}{\varphi_2\left(X^T\mathbf{y}(t), X^TB\right)} \\
\text{subject to} \quad & \eta_j\left(X^T\mathbf{y}(t), X^TB\right) \leq 0 \ \forall j \in \mathcal{J}_I, \\
& \eta_j\left(X^T\mathbf{y}(t), X^TB\right) = 0 \ \forall j \in \mathcal{J}_E,
\end{aligned} \quad (2)$$

where $\mathcal{J}_\mathcal{I}$ and $\mathcal{J}_\mathcal{E}$ denote the index sets for inequality and equality constraints respectively. Every function is considered to be real-valued. Note that X must always appear as an inner product with the signal $\mathbf{y}$ or with a deterministic $M \times L$ matrix B [16], which is also partitioned as $\mathbf{y}$ in (1):

$$B = [B_1^T, \ldots, B_k^T]^T \in \mathbb{R}^{M\times L}, \quad (3)$$

where each B_k is $M_k \times L$ and assumed to be available at node k. These deterministic matrices are independent of the time index t, and are often used to enforce a structure on the variable X (e.g. the constraint in the TRO example of Table 1 where $B = I_M$), or to formulate the problem in a deterministic framework without stochastic variables (e.g. least squares instead of minimum mean squared error). Additionally, the functions in (2) contain the stochastic signal $\mathbf{y}$ in their argument, which means they should contain an internal operator (such as an expected value) to extract a real-valued quantity from it. The ergodicity and short-term stationarity of $\mathbf{y}$ implies that a solution $X^*(t)$ of (2) at time sample t can be estimated using a window of observations of $\mathbf{y}$ around time point t. For mathematical tractability, we only consider time-independent solutions X^*, which corresponds to the assumption that the signal statistics of $\mathbf{y}$ do not change. In practice, the underlying signal statistics are allowed to change, assuming these dynamics are slower than the convergence speed of the algorithms we discuss, such that these algorithms are adaptively able to track changes in the statistics of the signals.

It is noted that Problem (2) can contain multiple variables (X), signals ($\mathbf{y}$) and deterministic matrices (B), even though only one of each is included in (2) for conciseness. For example, the TRO problem in Table 1 involves two $M-$channel signals ($\mathbf{y}$ and $\mathbf{v}$), and the RTLS example has two instances of B, $B_1 = I_M$ in the denominator of the cost function and $B_2 = L$ in the constraint.

3. FRACTIONAL PROGRAMMING REVIEW

A fractional program is an optimization problem with an objective function r represented by a ratio of two continuous and real-valued functions f_1 and f_2: $\min_{X\in\mathcal{S}} r(X) \triangleq f_1(X)/f_2(X)$. The constraint set $\mathcal{S} \subset \mathbb{R}^{M\times Q}$ is considered to be non-empty and compact and it is assumed that $f_2(X) > 0$ for $X \in \mathcal{S}$. We define the minimal value of r over $\mathcal{S}$ as $\rho^* \triangleq \min_{X\in\mathcal{S}} r(X)$ and the set of arguments achieving this value as $\mathcal{X}^* \triangleq \{X \in \mathcal{S} \mid r(X) = \rho^*\}$. To solve fractional programs, there exist generic methods based on solving auxiliary problems instead of the original problem [22]. The method

[1] This is the generic form of a DSFO problem as defined in [16], but for the special case where the cost function can be written as a ratio of two functions.

Algorithm 1: Dinkelbach's procedure [21]

X^0 initialized randomly, $\rho^0 \leftarrow r(X^0),\ i \leftarrow 0$
repeat
1) $X^{i+1} \leftarrow \underset{X\in\mathcal{S}}{\operatorname{argmin}}\ f_1(X) - \rho^i f_2(X)$.
2) $\rho^{i+1} \leftarrow r(X^{i+1}) = f_1(X^{i+1})/f_2(X^{i+1})$.
$i \leftarrow i+1$

we focus on in this paper is a parametric approach, initially presented in [21] and often referred to as Dinkelbach's procedure. Let us define the auxiliary functions $f(X,\rho) \triangleq f_1(X) - \rho f_2(X)$ and $g(\rho) \triangleq \min_{X\in\mathcal{S}} f(X,\rho)$, where ρ is a real scalar. Then, $g(\rho) = 0$ if and only if $\rho = \rho^*$ when $\mathcal{S}$ is compact [21, 23–25].

Dinkelbach's procedure is an iterative method aiming to find the unique root ρ^* of g, with a corresponding $X^* \in \mathcal{X}^*$, by iteratively solving the auxiliary problem $\min_{X\in\mathcal{S}}\ f(X,\rho^i) = f_1(X) - \rho^i f_2(X)$ as summarized in Algorithm 1. The convergence properties of Dinkelbach's procedure have been extensively studied in the literature. It can be shown that $(\rho^i)_i$ obtained from Algorithm 1 is a strictly decreasing sequence converging to ρ^* [24].

To solve DSFO problems with fractional objectives (2) in a centralized setting using Dinkelbach's procedure, the auxiliary problems corresponding to (2) are of the form

$$\underset{X\in\mathcal{S}}{\text{minimize}}\ \varphi_1\left(X^T\mathbf{y}(t), X^TB\right) - \rho^i\varphi_2\left(X^T\mathbf{y}(t), X^TB\right), \quad (4)$$

where we define $\mathcal{S}$ to be the constraint set of (2), assumed to be non-empty and compact and $\varphi_2 > 0$ for $X \in \mathcal{S}$.

4. MODIFYING DASF FOR FRACTIONAL PROGRAMS

The class of optimization problems written in the form (2) are a subclass of DSFO problems — namely those with a fractional objective — which can be solved in a fully distributed fashion using the DASF algorithm presented in [16]. However, solving Problem (2) by applying the DASF algorithm straightforwardly would lead to a high computational cost since it would require solving a full Dinkelbach procedure at each iteration, i.e., solving problems of the form (4) multiple times at each iteration of DASF. The F-DASF method we propose in this section interleaves the steps of the Dinkelbach procedure with the ones of the DASF algorithm to significantly reduce the computational cost at each node. A similar approach has been taken in [26] for a specific fractional program known as trace ratio optimization (see TRO in Table 1). The proposed F-DASF algorithm can therefore be viewed as a generalization of the algorithm in [26] towards generic fractional problems.

For the sake of completeness, we immediately define our F-DASF algorithm for networks with a general topology (we refer the reader to [16] for a more gentle introduction to DASF, which starts from simpler topologies). We define X^i to be the estimation of the global filter X at iteration i (with X^0 initialized randomly), partitioned as

$$X^i = [X_1^{iT}, \ldots, X_K^{iT}]^T, \quad (5)$$

where each X_k is $M_k \times Q$, such that $X^{iT}\mathbf{y} = \sum_k X_k^{iT}\mathbf{y}_k$ and $X^{iT}B = \sum_k X_k^{iT}B_k$. Each iteration starts with selecting an (arbitrary) updating node q. At each iteration i, the network is first pruned to obtain a tree $\mathcal{T}^i(\mathcal{G}, q)$ such that each pair of nodes is connected by a unique path. The pruning function can be chosen freely, as long as no link between the updating node q and its neighbors $n \in \mathcal{N}_q$ are removed [16], where $\mathcal{N}_q$ is to the set of neighboring nodes of node q. In the remaining parts of this section, the set of neighboring nodes of a certain node k corresponds to the one after pruning, with respect to $\mathcal{T}^i(\mathcal{G},q)$. Each node $k\in\mathcal{K}\backslash\{q\}$ compresses its M_k-channel signal $\mathbf{y}_k$ into a $Q-$channel signal using its current estimate X_k^i, while doing the same operation to its submatrix B_k to obtain

$$\widehat{\mathbf{y}}_k^i \triangleq X_k^{iT}\mathbf{y}_k,\ \widehat{B}_k^i \triangleq X_k^{iT}B_k. \tag{6}$$

In iteration i, the values in (6) are fused in an inwards flow within the tree $\mathcal{T}^i(\mathcal{G},q)$, to eventually arrive in the updating node q (this will be formalized later on). For $\widehat{\mathbf{y}}_k^i$, this means that a block of N samples is transmitted, where N should be large enough to estimate the relevant statistics from it [16]. The F-DASF algorithm will use different samples at each iteration, making the proposed method adaptive. The fusion flow emerges naturally within the tree based on the following rule. A node k waits until it has received data from all of its neighbors except one, say, node n. Node k will then fuse its local data (6) with the (fused/compressed) data received from the nodes $l\in\mathcal{N}_k\backslash\{n\}$, and transmits the result to node n. Formally, this means that node k will transmit to node n a batch of N samples of

$$\widehat{\mathbf{y}}_{k\to n}^i \triangleq X_k^{iT}\mathbf{y}_k+\sum_{l\in\mathcal{N}_k\backslash\{n\}}\widehat{\mathbf{y}}_{l\to k}^i, \tag{7}$$

where $\widehat{\mathbf{y}}_{l\to k}^i$ is the data received from its neighbor l. We observe that (7) is recursive in its second term, which vanishes for leaf nodes (nodes with a single neighbor). As a result, the recursion defined by (7) is initiated by all leafs of the tree. Eventually, node q will receive N samples of the fused and compressed signals

$$\widehat{\mathbf{y}}_{n\to q}^i = X_n^{iT}\mathbf{y}_n+\sum_{k\in\mathcal{N}_n\backslash\{q\}}\widehat{\mathbf{y}}_{k\to n}^i=\sum_{k\in\mathcal{B}_{nq}}\widehat{\mathbf{y}}_k^i \tag{8}$$

from all its neighbors $n\in\mathcal{N}_q$. The same fusion flow applies for the deterministic matrix B, using the $\widehat{B}_k^i$'s, such that node q receives $\widehat{B}_{n\to q}^i$, defined in a similar way to (8), from all its neighbors $n\in\mathcal{N}_q$. The set of nodes $\mathcal{B}_{nq}$ in (8) is defined as the subgraph of $\mathcal{T}^i(\mathcal{G},q)$ containing node n obtained after removing the link between nodes n and q. Defining $\mathcal{N}_q\triangleq\{n_1,\dots,n_{|\mathcal{N}_q|}\}$, the compressed signals gathered at node q and its own observation $\mathbf{y}_q$ can be structured as

$$\widetilde{\mathbf{y}}_q^i \triangleq [\mathbf{y}_q^T,\widehat{\mathbf{y}}_{n_1\to q}^{iT},\dots,\widehat{\mathbf{y}}_{n_{|\mathcal{N}_q|}\to q}^{iT}]^T\ \in\mathbb{R}^{\widetilde{M}_q}, \tag{9}$$

where $\widetilde{M}_q=|\mathcal{N}_q|\cdot Q+M_q$. Similarly, we can define an analogous quantity for the matrix B:

$$\widetilde{B}_q^i \triangleq [B_q^T,\widehat{B}_{n_1\to q}^{iT},\dots,\widehat{B}_{n_{|\mathcal{N}_q|}\to q}^{iT}]^T\ \in\mathbb{R}^{\widetilde{M}_q\times L}. \tag{10}$$

In the original DASF framework, node q would solve a compressed version of (2) based on the compressed data defined in (9) and (10), using the Dinkelbach procedure (Algorithm 1), which would converge to the global optimum under mild conditions [16]. However, instead of solving the full fractional problem, we propose that node q performs only a single iteration of Algorithm 1:

$$\begin{aligned}\underset{\widetilde{X}_q\in\mathbb{R}^{\widetilde{M}_q\times Q}}{\text{minimize}}\ &\varphi_1\left(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^T\widetilde{B}_q^i\right)-\rho^i\varphi_2\left(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^T\widetilde{B}_q^i\right)\\ \text{subject to}\ &\eta_j\left(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^T\widetilde{B}_q^i\right)\leq 0\ \ \forall j\in\mathcal{J}_I,\\ &\eta_j\left(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^T\widetilde{B}_q^i\right)=0\ \ \forall j\in\mathcal{J}_E,\end{aligned} \tag{11}$$

i.e., solves a compressed version of the auxiliary problem (4), where ρ^i can be computed as

$$\rho^i=\varrho\left(\widetilde{X}_q^{iT}\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^{iT}\widetilde{B}_q^i\right)=\frac{\varphi_1\big(\widetilde{X}_q^{iT}\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^{iT}\widetilde{B}_q^i\big)}{\varphi_2\big(\widetilde{X}_q^{iT}\widetilde{\mathbf{y}}_q^i(t),\widetilde{X}_q^{iT}\widetilde{B}_q^i\big)} \tag{12}$$

Algorithm 2: F-DASF Algorithm

X^0 initialized randomly, $i\leftarrow 0$.
repeat
  Choose the updating node as $q\leftarrow(i\mod K)+1$.
  1) The network $\mathcal{G}$ is pruned into a tree $\mathcal{T}^i(\mathcal{G},q)$.
  2) Every node k collects N samples of $\mathbf{y}_k$. All nodes compress these to N samples of $\widehat{\mathbf{y}}_k^i$ and also compute $\widehat{B}_k^i$ as in (6).
  3) The nodes sum-and-forward their compressed data towards node q via the recursive rule (7) (and a similar rule for the $\widehat{B}_k^i$'s). Node q eventually receives N samples of $\widehat{\mathbf{y}}_{n\to q}^i$ given in (8), and the matrix $\widehat{B}_{n\to q}^i$ defined similarly, from all its neighbors $n\in\mathcal{N}_q$.
  at *Node q* **do**
    4a) Compute ρ^i as in (12).
    4b) Compute a single Dinkelbach iteration by solving (11), resulting in $\widetilde{X}_q^{i+1}$. If the solution is not unique, select the solution which minimizes $||\widetilde{X}_q^{i+1}-\widetilde{X}_q^i||_F$ with $\widetilde{X}_q^i$ defined in (13).
    4c) Partition $\widetilde{X}_q^{i+1}$ as in (14).
    4d) Disseminate every G_n^{i+1} in the corresponding subgraph $\mathcal{B}_{nq}$.
  end
  5) Every node updates X_k^{i+1} according to (15).
  $i\leftarrow i+1$

at node q, with

$$\widetilde{X}_q^i=[X_q^{iT},I_Q,\dots,I_Q]^T. \tag{13}$$

Node q then obtains $\widetilde{X}_q^{i+1}$ by solving Problem (11). If the solution of (11) is not unique, $\widetilde{X}_q^{i+1}$ is selected such that it minimizes $||\widetilde{X}_q-\widetilde{X}_q^i||_F$ over all possible solutions $\widetilde{X}_q$ of (11), where $\widetilde{X}_q^i$ is given in (13). We partition $\widetilde{X}_q^{i+1}$ as

$$\widetilde{X}_q^{i+1}=[X_q^{(i+1)T},G_{n_1}^{(i+1)T},\dots,G_{n_{|\mathcal{N}_q|}}^{(i+1)T}]^T, \tag{14}$$

where G_n is $Q\times Q,\forall n\in\mathcal{N}_q$. Each G_n^{i+1} is then disseminated into the corresponding subgraph $\mathcal{B}_{nq}$ through node n so that every node can update its local variable estimator as

$$X_k^{i+1}=\begin{cases}X_q^{i+1} & \text{if } k=q\\ X_k^iG_n^{i+1} & \text{if } k\in\mathcal{B}_{nq},n\in\mathcal{N}_q.\end{cases} \tag{15}$$

Remark 1. It can be shown that each X^i, $i>0$, obtained from Algorithm 2 satisfies the constraints of the global problem (2), i.e., $X^i\in\mathcal{S}$ [16]. Additionally, a similar proof as in [16] shows that $X^i\in\mathcal{S}\iff\widetilde{X}_q^i\in\widetilde{\mathcal{S}}_q^i$ for every $i>0$, where $\widetilde{\mathcal{S}}_q^i$ is the constraint set of (11).

The entire process is then repeated by selecting other updating nodes at different iterations. The proposed fractional DASF (F-DASF) algorithm is summarized in Algorithm 2. We note that a new block of N samples of $\mathbf{y}$, say $\{\mathbf{y}(\tau)\}_{\tau=iN}^{(i+1)N-1}$, is used at each iteration i, hence the F-DASF algorithm acts as a block-adaptive filter which adapts to changes in the statistical properties of the measured signals. In particular, X^i is an estimator of $X^*(t)$ for $t=iN$.

The following theorem gives a convergence result of the objective values obtained from the F-DASF algorithm. After that, we introduce Theorem 2, which provides a stronger convergence result under the same mild conditions as those for the original DASF algorithm [16, 17], although its proof is omitted due to space constraints.

Theorem 1. *The sequence $(\rho^i)_i$ of objective values obtained by F-DASF is monotonically non-increasing and converges.*

Proof. In this proof, we denote the constraint set of (11) as $\widetilde{\mathcal{S}}_q^i$ and omit the matrix B for conciseness, as it is treated in a similar way to $\mathbf{y}$. Let us define the objective function of (11) as $\varphi\big(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t),\rho\big) \triangleq \varphi_1\big(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t)\big) - \rho\,\varphi_2\big(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t)\big)$. From the definition of ρ^i given in equation (12), note that we have $\varphi\big(\widetilde{X}_q^{iT}\widetilde{\mathbf{y}}_q^i(t),\rho^i\big) = 0$. Since $\widetilde{X}_q^{i+1}$ solves (11), we have that $\varphi\big(\widetilde{X}_q^{(i+1)T}\widetilde{\mathbf{y}}_q^i(t),\rho^i\big) \leq \varphi\big(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t),\rho^i\big)$ for any $\widetilde{X}_q \in \widetilde{\mathcal{S}}_q^i$. In particular, since $\widetilde{X}_q^i \in \widetilde{\mathcal{S}}_q^i$ (see Remark 1), we have $\varphi\big(\widetilde{X}_q^{(i+1)T}\widetilde{\mathbf{y}}_q^i(t),\rho^i\big) \leq \varphi\big(\widetilde{X}_q^{iT}\widetilde{\mathbf{y}}_q^i(t),\rho^i\big) = 0$. Rearranging the terms of $\varphi\big(\widetilde{X}_q^{(i+1)T}\widetilde{\mathbf{y}}_q^i(t),\rho^i\big)$, we obtain $\frac{\varphi_1\left(\widetilde{X}_q^{(i+1)T}\widetilde{\mathbf{y}}_q^i(t)\right)}{\varphi_2\left(\widetilde{X}_q^{(i+1)T}\widetilde{\mathbf{y}}_q^i(t)\right)} = \rho^{i+1} \leq \rho^i$. Therefore, the sequence $(\rho^i)_i$ is monotonic non-increasing and since it is lower bounded by ρ^*, it must converge. □

Theorem 2 (Proof Omitted)**.** *Suppose that Problem (4) for any feasible ρ^i satisfies the convergence conditions of the original DASF algorithm (see [16, 17]). Then the sequences $(\rho^i)_i$ and $(X^i)_i$ obtained by F-DASF also converge respectively to the global minimum ρ^* and to an optimal point $X^* \in \mathcal{X}^*$ of Problem (2).*

5. SIMULATIONS

We assess the performance of the F-DASF algorithm on the regularized total least squares (RTLS) problem [19, 20]

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^M} \; & \frac{\mathbb{E}[|\mathbf{x}^T\mathbf{y}(t)-d(t)|^2]}{1+\mathbf{x}^T\mathbf{x}} = \frac{\mathbf{x}^T R_{\mathbf{yy}}\mathbf{x} - 2\mathbf{x}^T\mathbf{r}_{\mathbf{y}d} + r_{dd}}{1+\mathbf{x}^T\mathbf{x}} \\ \text{s. t. } & ||\mathbf{x}^T L||^2 \leq 1, \end{aligned} \tag{16}$$

where we have $X = \mathbf{x} \in \mathbb{R}^M$, i.e., $Q = 1$, $R_{\mathbf{yy}} = \mathbb{E}[\mathbf{y}(t)\mathbf{y}^T(t)]$, $\mathbf{r}_{\mathbf{y}d} = \mathbb{E}[d(t)\mathbf{y}(t)]$ and $r_{dd} = \mathbb{E}[d^2(t)]$. The RTLS problem has applications in signal estimation tasks when both the observation and source signals are noisy [27–29]. Note that in (16), we have two deterministic matrices B, $B_1 = I_M$ and $B_2 = L$, where the former appears in the denominator of the objective: $\mathbf{x}^T\mathbf{x} = (\mathbf{x}^T I_M)\cdot(\mathbf{x}^T I_M)^T$. We first consider a stationary setting with $\mathbf{y}(t) = \mathbf{a}\cdot s(t) + \mathbf{n}(t)$, where each time sample of s is drawn from $\mathcal{N}(0, 0.5)$, and each entry of $\mathbf{n}$ is drawn from $\mathcal{N}(0, 0.1)$. Moreover, $d(t) = s(t)+w(t)$, where the time samples of w follow $\mathcal{N}(0, 0.01)$. L is a diagonal matrix where each element of the diagonal is drawn from $\mathcal{N}(1, 0.1)$ while the elements of $\mathbf{a} \in \mathbb{R}^M$ follow $\mathcal{N}(0, 0.2)$. At each iteration i, a batch of $N = 10^4$ samples of $\mathbf{y}$ and d is used to solve the RTLS problem, such that the relationship between i and t is $i = \lfloor t/N \rfloor$. At iteration i of Algorithm 2, the problem solved at node q is the compressed auxiliary problem (11):

$$\begin{aligned} \min_{\widetilde{\mathbf{x}}_q\in\mathbb{R}^{\widetilde{M}_q}} \; & \widetilde{\mathbf{x}}_q^T R^i_{\widetilde{\mathbf{y}}_q\widetilde{\mathbf{y}}_q}\widetilde{\mathbf{x}}_q - 2\widetilde{\mathbf{x}}_q^T\mathbf{r}^i_{\widetilde{\mathbf{y}}_q d} + r_{dd} - \rho^i(1+\widetilde{\mathbf{x}}_q^T\widetilde{I}_q^i\widetilde{I}_q^{iT}\widetilde{\mathbf{x}}_q) \\ \text{s. t. } & ||\widetilde{\mathbf{x}}_q^T\widetilde{L}_q^i||^2 \leq 1, \end{aligned} \tag{17}$$

with $R^i_{\widetilde{\mathbf{y}}_q\widetilde{\mathbf{y}}_q} = \mathbb{E}[\widetilde{\mathbf{y}}_q^i(t)\widetilde{\mathbf{y}}_q^{iT}(t)]$ and $\mathbf{r}^i_{\widetilde{\mathbf{y}}_q d} = \mathbb{E}[\widetilde{\mathbf{y}}_q^i(t)d(t)]$. The signal $\widetilde{\mathbf{y}}_q^i$ is defined in (9), while $\widetilde{I}_q^i$ and $\widetilde{L}_q^i$ are defined as $\widetilde{B}_q^i$ given in (10) when $B = I_M$ and $B = L$ respectively. In comparison, the DASF algorithm will require node q to solve a compressed version of (16), solved at each iteration by applying the Dinkelbach algorithm, i.e., by solving the auxiliary problem (17) multiple times. We measure the performance of the F-DASF algorithm compared to the DASF algorithm by looking both at the number of computations and the mean squared error (MSE) $\epsilon(\mathbf{x}^i) = ||\mathbf{x}^i - \mathbf{x}^*||^2 \cdot ||\mathbf{x}^*||^{-2}$,

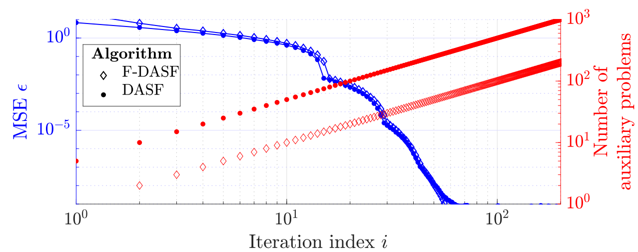

Fig. 1: MSE and cumulative computational cost across iterations of the DASF and the proposed F-DASF algorithm in a stationary setting.

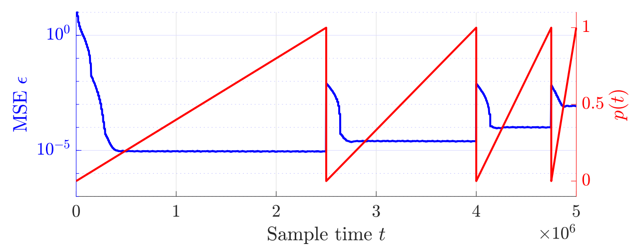

Fig. 2: MSE and p versus sample time t in an adaptive setting.

where $\mathbf{x}^*$ is the solution of (16). The stopping criterion we use for the Dinkelbach algorithm in each iteration of DASF is a minimum threshold of 10^{-10} on the norm of two consecutive $\widetilde{\mathbf{x}}_q$'s. Figure 1 shows the comparison of these performance metrics for a network with $K = 15$ nodes, each with $M_k = 5$ channels, with a randomly generated topology, and where the pruning function $\mathcal{T}^i(\cdot, q)$ is chosen to be the shortest path. Each point has been obtained by taking the median MSE over 100 Monte-Carlo runs. We see that using the DASF algorithm to solve the RTLS problem (16) in a distributed fashion requires a significantly higher number of computations compared to solving (16) using the F-DASF algorithm. In particular, the DASF algorithm requires on average solving 5 times more auxiliary problems per signal batch compared to the F-DASF, while still achieving an equivalent convergence speed.

Let us now consider the case where $\mathbf{a}$ changes at each time instance t, implying that the stationarity assumption on $\mathbf{y}$ does not hold anymore. We have $\mathbf{a}(t) = \mathbf{a}_0\cdot(1-p(t)) + (\mathbf{a}_0+\Delta)\cdot p(t)$, where p is represented in Figure 2 and the entries of $\mathbf{a}_0$ and Δ are drawn from $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 0.01)$ respectively. Figure 2 shows the MSE as a function of sample time t, where we see that the F-DASF algorithm is able to track slow changes in the signal statistics and correct abrupt ones, shown by sudden increase followed by gradual decrease in MSE values, highlighting its adaptive properties. Note that the algorithm reaches an MSE floor, rather than converging to 0 due to the fact that the target $\mathbf{x}^*$ changes at each iteration. The faster the rate of change in statistics, the higher the value of the MSE floor.

6. CONCLUSION

We have proposed an alternative method to the DASF algorithm for solving fractional spatial filtering problems in a distributed fashion over a network, and provided a proof of convergence in cost. The proposed F-DASF algorithm significantly reduces the computational cost while achieving the same convergence rate as the DASF method. In future work, we will extensively analyze the convergence properties of the F-DASF algorithm.

7. REFERENCES

[1] Simon Haykin and KJ Ray Liu, *Handbook on array processing and sensor networks*, John Wiley & Sons, 2010.

[2] Barry D Van Veen and Kevin M Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE assp magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[3] Alexander Bertrand, "Distributed signal processing for wireless EEG sensor networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 923–935, 2015.

[4] Benjamin Blankertz, Ryota Tomioka, Steven Lemm, Motoaki Kawanabe, and Klaus-Robert Muller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal processing magazine*, vol. 25, no. 1, pp. 41–56, 2007.

[5] Herbert Ramoser, Johannes Muller-Gerking, and Gert Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[6] Emil Björnson and Luca Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, 2020.

[7] Elina Nayebi, Alexei Ashikhmin, Thomas L Marzetta, and Bhaskar D Rao, "Performance of cell-free massive MIMO systems with MMSE and LSFD receivers," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 203–207.

[8] Jie Zhang, Richard Heusdens, and Richard Christian Hendriks, "Rate-distributed spatial filtering based noise reduction in wireless acoustic sensor networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2015–2026, 2018.

[9] Jacob Benesty, Jingdong Chen, and Yiteng Huang, *Microphone array signal processing*, vol. 1, Springer Science & Business Media, 2008.

[10] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[11] D Praveen Kumar, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, vol. 49, pp. 1–25, 2019.

[12] Jianshu Chen and Ali H Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[13] Reza Olfati-Saber and Jeff S Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 6698–6703.

[14] Cassio G Lopes and Ali H Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[15] Alexandros G Dimakis, Soummya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[16] Cem Ates Musluoglu and Alexander Bertrand, "A unified algorithmic framework for distributed adaptive signal and feature fusion problems–part I: Algorithm derivation," *arXiv preprint arXiv:2208.08867*, 2022.

[17] Cem Ates Musluoglu, Charles Hovine, and Alexander Bertrand, "A unified algorithmic framework for distributed adaptive signal and feature fusion problems–part II: Convergence properties," *arXiv preprint arXiv:2208.09088*, 2022.

[18] Huan Wang, Shuicheng Yan, Dong Xu, Xiaoou Tang, and Thomas Huang, "Trace ratio vs. ratio trace for dimensionality reduction," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[19] Diana M Sima, Sabine Van Huffel, and Gene H Golub, "Regularized total least squares based on quadratic eigenvalue problem solvers," *BIT Numerical Mathematics*, vol. 44, no. 4, pp. 793–812, 2004.

[20] Amir Beck, Aharon Ben-Tal, and Marc Teboulle, "Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 2, pp. 425–445, 2006.

[21] Werner Dinkelbach, "On nonlinear fractional programming," *Management science*, vol. 13, no. 7, pp. 492–498, 1967.

[22] Siegfried Schaible and Toshidide Ibaraki, "Fractional programming," *European journal of operational research*, vol. 12, no. 4, pp. 325–338, 1983.

[23] Raj Jagannathan, "On some properties of programming problems in parametric form pertaining to fractional programming," *Management Science*, vol. 12, no. 7, pp. 609–615, 1966.

[24] Siegfried Schaible, "Fractional programming. II, on Dinkelbach's algorithm," *Management science*, vol. 22, no. 8, pp. 868–873, 1976.

[25] Jean-Pierre Crouzeix and Jacques A Ferland, "Algorithms for generalized fractional programming," *Mathematical Programming*, vol. 52, no. 1, pp. 191–207, 1991.

[26] Cem Ates Musluoglu and Alexander Bertrand, "Distributed adaptive trace ratio optimization in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3653–3670, 2021.

[27] Hao Zhu, Geert Leus, and Georgios B Giannakis, "Sparse regularized total least squares for sensing applications," in *2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2010, pp. 1–5.

[28] Armin Pruessner and Dianne P O'Leary, "Blind deconvolution using a regularized structured total least norm algorithm," *SIAM journal on matrix analysis and applications*, vol. 24, no. 4, pp. 1018–1037, 2003.

[29] Nicolas H Younan and X Fan, "Signal restoration via the regularized constrained total least squares," *Signal Processing*, vol. 71, no. 1, pp. 85–93, 1998.