An Online Parallel Algorithm for Spectrum Sensing in Cognitive Radio Networks

Yang Yang*, Mengyi Zhang†, Marius Pesavento*, Daniel P. Palomar‡
*Communication Systems Group, Darmstadt University of Technology. Email: {yang, pesavento}@nt.tu-darmstadt.de
†Dept. of Computer Science and Engineering, The Chinese Univ. of Hong Kong. Email: zhangmy@cse.cuhk.edu.hk
‡Dept. of Electronic and Computer Engineering, The Hong Kong Univ. of Science & Technology. Email: palomar@ust.hk

***Abstract*—We consider the estimation of the position and transmit power of primary users in cognitive radio networks based on solving a sequence of ℓ_1-regularized least-square problems, in which the unknown vector is sparse and the measurements are only sequentially available. We propose an online parallel algorithm that is novel in three aspects: i) all elements of the unknown vector variable are updated in parallel; ii) the update of each element has a closed-form expression; and iii) the stepsize is designed to accelerate the convergence yet it still has a closed-form expression. The convergence property is both theoretically analyzed and numerically consolidated.**

I. Introduction

Cognitive radio (CR) has been widely accepted as a enabling technique for flexible and efficient use of the radio spectrum [1], since it allows the unlicensed secondary users (SUs) to access the spectrum provided that the licensed primary users (PUs) are idle or the interference generated by the SUs to the PUs is below a level that is tolerable for the PUs [2], [3].

One prerequisite to apply CR is the ability to obtain an estimate of the position and transmit power of PUs so that the SUs can avoid the areas in which the PUs are actively transmitting [4], [5]. As a fundamental methodology for estimation, the minimum mean-square-error (MMSE) criterion has a solid root [6] and is adopted in a number of works [7], [8].

MMSE approach involves the calculation of the expectation of a least-square (LS) function that depends on the so-called regression vector and measurement output, both of which are random variables. When the statistics of these random variables are unknown, it is impossible to calculate the expectation analytically. An alternative is to use the sample average function as an approximation of the expectation, and this leads to the well-known recursive least-square (RLS) algorithm [6].

In practice, the signal to be estimated may be sparse in nature [4], [7], [8]. In a recent attempt to apply the RLS approach to estimate a sparse signal, a sparsity regularization function in terms of ℓ_1-norm is incorporated into the LS function in each iteration [7], [8], leading to a sparsity-regularized LS problem which has the form of least-absolute shrinkage and selection operator (LASSO).

However, a closed-form solution to the ℓ_1-regularized LS problem no longer exists because of the ℓ_1-norm regularization function and the problem can only be solved iteratively [9]. Since the observations are sequentially available, and with each observation, a ℓ_1-regularized LS problem is formed and solved, the overall complexity of using solvers for the sequence of ℓ_1-regularized LS problems is no longer affordable.

To reduce the complexity, an *online* sequential algorithm is proposed in [7]: the ℓ_1-regularized LS problem is solved with respect to (w.r.t.) only a single element of the unknown vector variable (instead of *all* elements as in a solver) while other elements are fixed, and the element update has a closed-form expression based on the soft-thresholding operator. After a new sample arrives, a new l_1-regularized LS problem is formed and solved, but only w.r.t. the next element.

Intuitively, since the elements are updated sequentially and only a single element is updated at each time instance, the online algorithm proposed in [7] sometimes suffers from slow convergence. It is tempting to use the parallel algorithm proposed in [10], [11], but it converges for deterministic optimization problems only. Besides, its convergence speed heavily depends on the decay rate of the diminishing stepsize: on the one hand, a slowly decaying stepsize is preferable to make notable progress in each iteration and to achieve satisfactory convergence speed; on the other hand, theoretical convergence is guaranteed only when the stepsize decays fast enough. In practice, it is a difficult task on its own to find the decay rate that gives the optimal trade-off.

In this paper, we propose an online parallel algorithm with provable convergence for recursive estimation of sparse signals. In particular, our contributions are as follows:

1) At each time instance, all elements are updated in parallel, and the convergence speed is thus greatly enhanced compared with [7]. As a nontrivial extension of [7] from sequential update to parallel update and [10], [11] from deterministic optimization problems to stochastic optimization problems, we rigorously show that the proposed algorithm almost surely converges.

2) The proposed stepsize is based on the so-called minimization rule, so notable progress is achieved all the time and the trouble of parameter tuning in [10], [11] is saved. Besides, both the update of each element and the stepsize can be calculated in closed-form, so the algorithm is very easy to implement and fast to converge.

3) The proposed algorithm can be implemented in both a centralized manner and, if necessary, a distributed manner. In the latter case, the signaling is much less than in state-of-the-art techniques [4], [8].

978-1-4799-8297-4/14/$31.00 ©2014 IEEE

II. System Model and Problem Formulation

Consider a CR network composed of an unknown number of PUs and N SUs. The PUs and SUs are located in a two-dimensional geographical area $\mathcal{A}$, and the positions of the PUs are unknown to the SUs. To locate the PUs, we discretize the geographical area $\mathcal{A}$ into K grid points, with $g_{n,k}$ being the channel gain from position k to SU n, and x_k being the transmit power of PUs located at position k. With non-coherent energy detectors, y_n is the received power measured by SU n:

$$y_n = \langle \mathbf{g}_n, \mathbf{x}^\star \rangle + v_n, \quad n = 1, \ldots, N, \tag{1}$$

where $\mathbf{g}_n \triangleq \{g_{n,k}\}_{k=1}^K \in \mathbb{R}^K$, $\mathbf{x}^\star \triangleq \{x_k^\star\}_{k=1}^K \in \mathbb{R}^K$, and $v_n \in \mathbb{R}$ is the additive estimation noise. Throughout the paper, we make the following blanket assumptions on $\mathbf{g}_n$ and v_n:

(A1) $\mathbf{g}_n$ are i.i.d. random variables with a bounded positive definite covariance matrix;

(A2) v_n are i.i.d. random variables with zero mean and bounded variance, and it is uncorrelated with $\mathbf{g}_n$.

We assume $\mathbf{g}_n$ is known at SU n; in practice, it can be estimated by SU n using Kriged Kalman filtering as proposed in [4], which is based on an interpolation process given the training data sent by other SUs only. A detailed description is out of the scope of this paper, but we remark that 1) estimating $\mathbf{g}_n$ does not require the cooperation of PUs, and 2) its complexity is independent of the number of grid points K but depends only on the number of SUs N.

Given the linear model in (1), the problem is to estimate $\mathbf{x}^\star$ from the set of measurement and regression vector $\{\mathbf{g}_n, y_n\}_{n=1}^N$. Since both the regression vector $\mathbf{g}_n$ and estimation noise v_n are random variables, the received signal y_n is also random. A fundamental approach to estimate $\mathbf{x}^\star$ is based on the so-called minimum mean-square-error (MMSE) criterion, and this MMSE approach has a solid root in adaptive filter theory [6]. Note that it is advisory to estimate $\mathbf{x}$ jointly from all of the measurements $\{\mathbf{g}_n, y_n\}_{n=1}^N$ to overcome the so-called hidden node problem, since some of the SUs may be in shadowed area and they may miss the presence of a PU based on their own measurements only [5], [8], [12]. Given the above statements, the cooperative estimation problem is formulated as follows:

$$\mathbf{x}^\star = \underset{\mathbf{x}=(x_k)_{k=1}^K}{\arg\min}\ \mathbb{E}\left[\sum_{n=1}^N \left(y_n - \langle \mathbf{g}_n, \mathbf{x}\rangle\right)^2\right], \tag{2}$$

where the expectation is over $\{\mathbf{g}_n, y_n\}_{n=1}^N$.

In practice, the statistics of $\{\mathbf{g}_n, y_n\}_{n=1}^N$ are not necessarily available to compute the expectation in (2) analytically, but the samples of $\{\mathbf{g}_n, y_n\}_{n=1}^N$ are much easier to obtain, and one alternative is to approximate the expectation in (2) by the sample average function constructed from the samples $\{\mathbf{g}_n^{(\tau)}, y_n^{(\tau)}\}_{\tau=1}^t$ sequentially available up to time t [6]:

$$\begin{aligned}\mathbf{x}_{\text{rls}}^{(t)} &\triangleq \arg\min_{\mathbf{x}} \frac{1}{t}\sum_{\tau=1}^t \sum_{n=1}^N \left(y_n^{(\tau)} - \langle \mathbf{g}_n^{(\tau)}, \mathbf{x}\rangle\right)^2 \\ &= \arg\min_{\mathbf{x}} \frac{1}{2}\langle \mathbf{x}, \mathbf{G}^{(t)}\mathbf{x}\rangle - \langle \mathbf{b}^{(t)}, \mathbf{x}\rangle,\end{aligned} \tag{3}$$

where

$$\mathbf{G}^{(t)} \triangleq \frac{1}{t}\sum_{\tau=1}^t \sum_{n=1}^N \mathbf{g}_n^{(\tau)}(\mathbf{g}_n^{(\tau)})^T,\ \mathbf{b}^{(t)} \triangleq \frac{1}{t}\sum_{\tau=1}^t \sum_{n=1}^N y_n^{(\tau)}\mathbf{g}_n^{(\tau)}. \tag{4}$$

In literature, (3) is known as RLS, and $\mathbf{x}_{\text{rls}}^t$ is a strongly consistent estimator of $\mathbf{x}^\star$, i.e., $\lim_{t\to\infty} \mathbf{x}_{\text{rls}}^{(t)} = \mathbf{x}^\star$, almost surely, under Assumptions (A1)-(A2) [7].

If the unknown signal $\mathbf{x}^\star$ is furthermore sparse by nature or by design, $\mathbf{x}_{\text{rls}}^{(t)}$ given by (3) may not be a good estimate because it is not necessarily sparse, unless when t is sufficiently large. To overcome this shortcoming, a sparsity encouraging function in terms of ℓ_1-norm is incorporated into (3), leading to the ℓ_1-regularized loss function [7]:

$$L^{(t)}(\mathbf{x}) \triangleq \frac{1}{2}\langle \mathbf{x}, \mathbf{G}^{(t)}\mathbf{x}\rangle - \langle \mathbf{b}^{(t)}, \mathbf{x}\rangle + \mu^{(t)} \left\|\mathbf{x}\right\|_1, \tag{5}$$

where $\mu^{(t)} > 0$. Then by minimizing the sample average function $L^{(t)}(\mathbf{x})$, we obtain the estimate $\mathbf{x}_{\text{lasso}}^{(t)}$ given by

$$\mathbf{x}_{\text{lasso}}^{(t)} \triangleq \arg\min_{\mathbf{x}} L^{(t)}(\mathbf{x}), \tag{6}$$

In literature, problem (6) for a fixed t is known as the *least-absolute shrinkage and selection operator* (LASSO).

Compared with (2) whose objective function is stochastic and whose calculation depends on unknown parameters, (6) is a well-defined deterministic optimization problem whose theoretic and algorithmic properties are well investigated and understood. The connection between $\mathbf{x}_{\text{lasso}}^{(t)}$ in (6) and the unknown variable $\mathbf{x}^\star$ is given in the following lemma [7].

Lemma 1. *Suppose Assumptions (A1)-(A2) as well as the following assumption are satisfied for (6):*

(A3) $\{\mu^{(t)}\}$ *is a positive sequence converging to* 0*, i.e.,* $\mu^{(t)} > 0$ *and* $\lim_{t\to\infty} \mu^{(t)} = 0$.

Then $\lim_{t\to\infty} \mathbf{x}_{lasso}^{(t)} = \mathbf{x}^\star$ *almost surely.*

Note that $\mathbf{x}^\star$ in (1) is the power vector and thus always nonnegative by definition. However, to make the proposed algorithm applicable in an even broader context, we do not make this assumption. We will show later how this nonnegative property can be exploited to further strengthen the results.

III. The Online Parallel Algorithm

Lemma 1 not only states the connection between $\mathbf{x}_{\text{lasso}}^{(t)}$ and $\mathbf{x}^\star$ from a theoretical perspective, but also offers valuable insights on the estimation of $\mathbf{x}^\star$ from the algorithmic point of view: it is clear that the recursive estimation of the unknown signal $\mathbf{x}^\star$ is achieved by solving a sequence of deterministic optimization problems, each of which has the form (6) and can be solved by numerous solvers, e.g., `FISTA` [9]. Actually, this methodology has been adopted in a number of works [8].

However, solving (6) completely w.r.t. all elements of $\mathbf{x}$ at each time instance is computationally impractical. To reduce the complexity, an "online" recursive algorithm is proposed in [7]: at each iteration t, (6) is only solved approximately. In particular, only a single element x_k with $k = \text{mod}(t-1, K)+1$

is updated by minimizing $L^{(t)}(\mathbf{x})$ w.r.t. x_k only while the remaining elements $\{x_j\}_{j\neq k}$ are assumed to be fixed; then in the next iteration $t+1$, a new sample average function $L^{(t+1)}(\mathbf{x})$ is constructed with the newly arrived samples, and the $(k+1)$-element is updated by minimizing $L^{(t+1)}(\mathbf{x})$ w.r.t. x_{k+1} only, while the remaining elements again are fixed. Although it is easy to implement, this sequential update scheme suffers from slow convergence, and the incurred delay is even larger when K is large.

To overcome the slow convergence, we propose an online parallel update scheme in which all elements are updated simultaneously. At the t-th iteration of the proposed algorithm, when updating x_k where $1 \leq k \leq K$, we follow a two-step procedure. In the first step, the update direction of x_k at $x_k = x_k^{(t)}$, denoted as $\hat{x}_k^{(t)} - x_k^{(t)}$, is determined based on the so-called nonlinear best-response, i.e., $\hat{x}_k^{(t)}$ is given by:

$$\hat{x}_k^{(t)} \triangleq \arg\min_{x_k} \Big\{ L^{(t)}(x_k, \mathbf{x}_{-k}^{(t)}) + \frac{1}{2} c_k^{(t)} (x_k - x_k^{(t)})^2 \Big\}, \quad \forall\, k, \tag{7}$$

where $\mathbf{x}_{-k} \triangleq \{x_j\}_{j\neq k}$ and an additional quadratic proximal term with $c_k^{(t)} \geq 0$ is included for numerical simplicity and stability [10], [13]. Note that in (7), remaining elements $\mathbf{x}_{-k}$ are fixed to their values of the preceding iteration $\mathbf{x}_{-k} = \mathbf{x}_{-k}^{(t)}$. Then in the second step, given the update direction $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$, an intermediate variable $\tilde{\mathbf{x}}^{(t+1)}$ is defined according to:

$$\tilde{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} + \gamma^{(t)} (\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}), \tag{8}$$

where $\gamma^{(t)} \in [0,1]$ is the stepsize. The intermediate variable $\tilde{\mathbf{x}}^{(t+1)}$ plays an important role in the construction of $\mathbf{x}^{(t+1)}$, and their connection, as well as the selection of the stepsize $\gamma^{(t+1)}$, will be discussed shortly later.

In view of the analytical expression of $L^{(t)}(\mathbf{x})$ in (5), the best-response defined in (7) can be expressed in closed-form:

$$\begin{aligned} \hat{x}_k^{(t)} &= \arg\min_{x_k} \left\{ \begin{array}{l} \frac{1}{2} G_{kk}^{(t)} x_k^2 + r_k^{(t)} \cdot x_k \\ + \mu^{(t)} |x_k| + \frac{1}{2} c_k^{(t)} (x_k - x_k^{(t)})^2 \end{array} \right\} \\ &= \frac{1}{G_{kk}^{(t)} + c_k^{(t)}} \mathcal{S}_{\mu^{(t)}} (r_k^{(t)} - c_k^{(t)} x_k^{(t)}), \end{aligned} \tag{9}$$

where $\mathcal{S}_a(b) \triangleq [-b-a]^+ - [b-a]^+$ is the well-known soft-thresholding operator [9], $\mathbf{r}^{(t)} \triangleq \mathbf{G}^{(t)}\mathbf{x}^{(t)} - \text{diag}(\mathbf{G}^{(t)})\mathbf{x}^{(t)} - \mathbf{b}^{(t)}$, and $\text{diag}(\mathbf{X})$ is a diagonal matrix whose diagonal elements are obtained from $\mathbf{X}$. Besides, since $\mathbf{G}^{(t)} \succeq \mathbf{0}$ and $G_{kk}^{(t)} \geq 0$, $c_k^{(t)}$ should be selected such that $G_{kk}^{(t)} + c_k^{(t)} > 0$.

The update direction $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$ is a descent direction of $L^{(t)}(\mathbf{x})$ in the sense specified by the following proposition, whose proof is omitted due to page limit [14].

Proposition 2. *For the update direction $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$ while $\hat{\mathbf{x}}^{(t)}$ is given in (9), the following holds for any $\gamma \in [0,1]$:*

$$\begin{aligned} L^{(t)}(\mathbf{x}^{(t)} + &\gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) - L^{(t)}(\mathbf{x}^{(t)}) \leq \\ &-\gamma \left(c_{min}^{(t)} - \frac{1}{2} \lambda_{\max}(\mathbf{G}^{(t)}) \gamma \right) \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2^2, \end{aligned} \tag{10}$$

where $c_{\min}^{(t)} \triangleq \min_k \left\{ G_{kk}^{(t)} + c_k^{(t)} \right\} > 0$.

Now we discuss how to select the stepsize $\gamma^{(t)}$ so that fast convergence is observed. As shown in Proposition 2, $L^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) < L^{(t)}(\mathbf{x}^t)$ when γ is sufficiently small, so one natural choice of the stepsize rule is the so-called "minimization rule" [15, Sec. 2.2.1]:

$$\begin{aligned} \gamma^{(t)} &= \arg\min_{0\leq\gamma\leq 1} L^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) \\ &= \arg\min_{0\leq\gamma\leq 1} \left\{ \begin{array}{l} \frac{1}{2} \big\langle \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}, \mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \big\rangle \cdot \gamma^2 \\ + \big\langle \mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)}, \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \big\rangle \cdot \gamma \\ + \mu^{(t)} \left\| \mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \right\|_1 \end{array} \right\}. \end{aligned} \tag{11}$$

That is, the stepsize is selected such that the objective value is decreased to the largest extent: for any $\gamma \in [0,1]$,

$$L^{(t)}(\mathbf{x}^{(t)} + \gamma^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) \leq L^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})). \tag{12}$$

But the difficulty with this minimization rule is the complexity of solving (11), since the presence of the ℓ_1 makes it impossible to find a closed-form solution and (11) can only be solved numerically by a solver such as `MOSEK` [16].

To find a stepsize which yields fast convergence but is much easier to calculate, we propose a simplified minimization rule based on the convexity of vector norms. To see the insight, we split $L^{(t)}(\mathbf{x})$ defined in (5) into a smooth part $f^{(t)}(\mathbf{x})$ and a nonsmooth part $h^{(t)}(\mathbf{x})$:

$$f^{(t)}(\mathbf{x}) \triangleq \frac{1}{2} \big\langle \mathbf{x}, \mathbf{G}^{(t)}\mathbf{x} \big\rangle - \big\langle \mathbf{b}^{(t)}, \mathbf{x} \big\rangle, \tag{13a}$$

$$h^{(t)}(\mathbf{x}) \triangleq \mu^{(t)} \left\| \mathbf{x} \right\|_1. \tag{13b}$$

It follows from the convexity of $h^t(\mathbf{x})$ that for any $\gamma \in [0,1]$:

$$\begin{aligned} h^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) &= \\ &\leq (1-\gamma) h^t(\mathbf{x}^{(t)}) + \gamma h^t(\hat{\mathbf{x}}^{(t)}) \\ &= h^{(t)}(\mathbf{x}^{(t)}) + \gamma(h^{(t)}(\hat{\mathbf{x}}^{(t)}) - h^{(t)}(\mathbf{x}^{(t)})), \end{aligned} \tag{14}$$

while the right hand side of (14) is linear in γ, and equality is achieved either when $\gamma = 0$ or $\gamma = 1$. As a result, the function

$$\begin{aligned} \bar{L}^{(t)}(\gamma) \triangleq\, & f^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) \\ & + \gamma(h^{(t)}(\hat{\mathbf{x}}^{(t)}) - h^{(t)}(\mathbf{x}^{(t)})) + h^{(t)}(\mathbf{x}^{(t)}) \end{aligned} \tag{15}$$

is a tight upper bound of $L^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}))$.

Then in the simplified minimization rule, instead of directly minimizing $L^{(t)}(\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}))$ over γ, we minimize its upper bound $\bar{L}^{(t)}(\gamma)$ and $\gamma^{(t)}$ is accordingly given by

$$\begin{aligned} \gamma^{(t)} &= \arg\min_{0\leq\gamma\leq 1} \bar{L}^{(t)}(\gamma) \\ &= \arg\min_{0\leq\gamma\leq 1} \left\{ \begin{array}{l} \frac{1}{2} \big\langle \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}, \mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \big\rangle \cdot \gamma^2 \\ + \big\langle (\mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)}, \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \big\rangle \cdot \gamma \\ + \mu^{(t)} (\left\| \hat{\mathbf{x}}^{(t)} \right\|_1 - \left\| \mathbf{x}^{(t)} \right\|_1) \cdot \gamma \end{array} \right\}. \end{aligned} \tag{16}$$

The scalar problem (16) is convex quadratic with a bound constraint and it has a closed-form solution given by (17) at the top of the next page, where $[x]_0^1 \triangleq \min(\max(x,0),1)$. At

$$\gamma^{(t)} = \left[-\frac{\left\langle \mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)}, \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\rangle + \mu^{(t)}\left(\left\|\hat{\mathbf{x}}^{(t)}\right\|_1 - \left\|\mathbf{x}^{(t)}\right\|_1\right)}{\left\langle \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}, \mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \right\rangle} \right]_0^1 \tag{17}$$

Algorithm 1: The Online Parallel Algorithm

```
Data: $\mathbf{x}^{(0)} = \mathbf{0}$; set $t = 0$.
Step 1: If $\mathbf{x}^{(t)}$ satisfies a suitable termination criterion: STOP.
Step 2: Calculate $\tilde{\mathbf{x}}^{(t+1)}$ according to
        $\tilde{\mathbf{x}}^{(t+1)} = (1 - \gamma^{(t)})\mathbf{x}^{(t)} + \gamma^{(t)}\hat{\mathbf{x}}^{(t)}$,
    where $\hat{\mathbf{x}}^{(t)}$ and $\gamma^{(t)}$ is determined according to (9) and (17), respectively.
Step 3: If $L^{(t)}(\tilde{\mathbf{x}}^{(t+1)}) \leq 0$, $\mathbf{x}^{(t+1)} = \tilde{\mathbf{x}}^{(t+1)}$; otherwise $\mathbf{x}^{(t+1)} = \mathbf{0}$.
Step 4: $t \leftarrow t + 1$ and go back to Step 1.
```

the same time, it can also yield a strict decrease in $L^{(t)}(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^{(t)}$ as the standard minimization rule does in (12). We can see this from the following inequalities:

$$\begin{aligned} L^{(t)}(\tilde{\mathbf{x}}^{(t+1)}) &= L^{(t)}(\mathbf{x}^{(t)} + \gamma^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) \\ &\leq \bar{L}^{(t)}(\gamma^{(t)}) < \bar{L}^{(t)}(\gamma)\big|_{\gamma=0} = L^{(t)}(\mathbf{x}^{(t)}), \end{aligned}$$

where the first inequality comes from the convexity of $h^{(t)}(\mathbf{x})$ as in (14), and the second strict inequality comes from the uniqueness of the optimal solution of (16) and the fact that $\gamma^{(t)} \neq 0$.

Furthermore, it is sometimes possible to obtain an estimate with an even smaller objective value than $L^{(t)}(\tilde{\mathbf{x}}^{(t+1)})$. A look at the definition of $L^{(t)}(\mathbf{x})$ in (5) reveals that $L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}) \leq L^{(t)}(\mathbf{0}) = 0$ for all t, because $\mathbf{x}_{\text{lasso}}^{(t)}$ defined in (6) is the minimizing variable of $L^{(t)}(\mathbf{x})$ while $\mathbf{x} = \mathbf{0}$ is just a feasible solution. Depending on the value of $L^{(t)}(\tilde{\mathbf{x}}^{(t+1)})$, we update $\mathbf{x}$ as follows:

$$\mathbf{x}^{(t+1)} = \begin{cases} \tilde{\mathbf{x}}^{(t+1)}, & \text{if } L^{(t)}(\tilde{\mathbf{x}}^{(t+1)}) \leq 0, \\ \mathbf{0}, & \text{otherwise}. \end{cases} \tag{18}$$

To summarize the above analysis, the proposed online parallel algorithm is formally described in Algorithm 1, and its convergence properties are given in the following theorem, whose proof is omitted due to page limit [14].

Theorem 3. *Suppose Assumptions (A1)-(A3) as well as the following assumptions are satisfied:*

(A4) *Both $\mathbf{g}_n$ and v_n have bounded moments;*
(A5) *$G_{kk}^{(t)} + c_k^{(t)} \geq c$ for some $c > 0$;*
(A6) *The sequence $\{\mu^{(t)}\}$ is decreasing, i.e., $\mu^{(t+1)} \leq \mu^{(t)}$.*

Then the sequence $\left\{\mathbf{x}^{(t)}\right\}_t$ produced by Algorithm 1 converges to $\mathbf{x}^{\star}$ almost surely.

Assumption (A4) is a standard assumption on random variables and can usually be satisfied in practice. Assumption (A5) is satisfied if $c_k^{(t)}$ is lower bounded by some positive scalar for all t. As for Assumption (A6), it is satisfied by the previously mentioned choices of $\mu^{(t)}$, e.g., $\mu^{(t)} = \alpha/t^{\beta}$ with $\alpha > 0$ and $\beta > 0$. Typical choices of β are β =0.5 and 1 [7].

In what follows, we comment on some of the novel features of Algorithm 1 that make it appealing in practice:

1) Algorithm 1 is an instance of online algorithms where problem (6) is solved only approximately. Compared with [8] where (6) is solved exactly in each iteration, the complexity is greatly reduced without jeopardizing the convergence property.

2) Algorithm 1 is an instance of parallel algorithms where all elements are updated simultaneously in each iteration. Compared with sequential algorithms where only one element can be updated while the others have to remain fixed [7], the improvement in convergence speed is presumably notable, especially when the signal dimension is large.

3) The implementation of Algorithm 1 is very easy, since both the computations of the best-response and the stepsize have closed-form expressions. With the proposed stepsize rule, notable decrease in objective function value is guaranteed in all iterations, and this saves the trouble of tuning the free parameters as required in [10].

4) Algorithm 1 converges under milder assumptions than state-of-the-art. For example, we do not require the regression vector $\mathbf{g}_n$ and noise v_n to be uniformly bounded, which was however assumed in [17] but cannot be satisfied if they follow, e.g., the Gaussian distribution.

Note that Algorithm 1 can be implemented in a distributed manner among SUs with limited signaling. We omit the details due to page limit and interested readers are referred to [14] for more details.

A. Estimation of the nonnegative power vector

Indeed, the unknown vector $\mathbf{x}^{\star}$ in (1) represents the transmit power of PUs which is always nonnegative, and the analysis can thus be simplified. Specifically, the best-response $\hat{x}_k^{(t)}$ in (9) is simplified to

$$\hat{x}_k^{(t)} = \frac{\left[-(r_k^{(t)} - h_k^{(t)}x_k^{(t)}) - \mu^{(t)}\right]^+}{G_{kk}^{(t)} + h_k^{(t)}}.$$

Furthermore, since both $\mathbf{x}^{(t)}$ and $\hat{\mathbf{x}}^{(t)}$ are nonnegative, we have $\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \geq \mathbf{0}$ for $\gamma \in [0, 1]$ and $\left\|\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})\right\|_1 = \sum_{k=1}^{K} x_k^{(t)} + \gamma(\hat{x}_k^{(t)} - x_k^{(t)})$. Therefore one can directly adopt the minimization rule and the stepsize is accordingly given as

$$\gamma^{(t)} = \left[-\frac{\left\langle \mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)} + \mu^{(t)}\mathbf{1}, \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\rangle}{\left\langle \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}, \mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \right\rangle} \right]_0^1,$$

where $\mathbf{1}$ is a vector with all elements equal to 1.

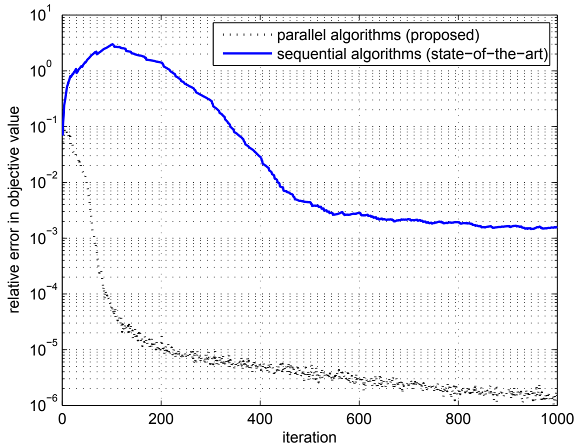

Figure 1. Convergence behavior in terms of objective function value

IV. Numerical Results

In this section, we test the convergence behavior of Algorithm 1 with the online sequential algorithm proposed in [7]. In this example, the parameters are selected as follows:

- $N = 1$, so the subscript n is omitted.
- the dimension of $\mathbf{x}^{\star}$: $K = 100$;
- the density of $\mathbf{x}^{\star}$: 0.1;
- Both $\mathbf{g}$ and v are generated by i.i.d. standard normal distributions: $\mathbf{g} \in \mathcal{CN}(\mathbf{0}, \mathbf{I})$ and $v \in \mathcal{CN}(0, 0.2)$;
- The sparsity regularization gain $\mu^{(t)} = \sqrt{K}/t = 10/t$;
- Unless otherwise stated, the simulations results are averaged over 100 realizations.

We plot in Figure 1 the iteration t versus the relative error in objective value $(L^{(t)}(\mathbf{x}^{(t)}) - L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}))/L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)})$, where 1) $\mathbf{x}_{\text{lasso}}^{(t)}$ is defined in (6) and calculated by MOSEK [16]; 2) $\mathbf{x}^{(t)}$ is returned by Algorithm 1 in the proposed online parallel algorithm; 3) $\mathbf{x}^{(t)}$ is returned by [7, Algorithm 1] in online sequential algorithm; and 4) $\mathbf{x}^{(0)} = \mathbf{0}$ for both parallel and sequential algorithms. Note that $L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)})$ is by definition the lower bound of $L^{(t)}(\mathbf{x})$ and $L^{(t)}(\mathbf{x}^{(t)}) - L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}) \geq 0$ for all t. From Figure 1 it is clear that the proposed algorithm converges to a precision of 10^{-2} in less than 200 iterations while the sequential algorithm needs more than 800 iterations. The improvement in convergence speed is thus notable. If one sets the precision as 10^{-4}, the online sequential algorithm does not even converge in a reasonable number of iterations. Therefore, the proposed online parallel algorithm outperforms in both convergence speed and solution quality.

V. Conclusions

In this paper, we have proposed an online algorithm with provable convergence for the recursive estimation of sparse signals. Since all elements are updated in parallel and in closed-form, the convergence speed is greatly enhanced. The proposed simplified minimization stepsize rule makes notable progress possible in all iterations while it can still be calculated in closed-form, achieving a good trade-off between complexity and performance. The fast convergence speed of the proposed algorithm is also consolidated numerically.

Acknowledgment

The work of Y. Yang and M. Pesavento is supported by ADEL, the Seventh Framework Programme for Research of the European Commission under grant number 619647. The work of D. P. Palomar is supported by the Hong Kong RGC 617312 research grant.

References

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[2] R. Zhang, Y.-C. Liang, and S. Cui, "Dynamic Resource Allocation in Cognitive Radio Networks," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 102–114, May 2010.

[3] Y. Yang, G. Scutari, P. Song, and D. P. Palomar, "Robust MIMO Cognitive Radio Systems Under Interference Temperature Constraints," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2465–2482, Nov. 2013.

[4] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative Spectrum Sensing for Cognitive Radios Using Kriged Kalman Filtering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 24–36, Feb. 2011.

[5] F. Zeng, C. Li, and Z. Tian, "Distributed Compressive Spectrum Sensing in Cooperative Multihop Cognitive Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 37–48, Feb. 2011.

[6] A. Sayed, *Adaptive filters*. Hoboken, N.J.: Wiley-Interscience, 2008.

[7] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online Adaptive Estimation of Sparse Signals: Where RLS Meets the ℓ_1-Norm," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3436–3447, Jul. 2010.

[8] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," in *Academic Press Library in Signal Processing*, R. Chellappa and S. Theodoridis, Eds., 2014, vol. 2, pp. 329–408.

[9] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[10] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by Partial Linearization: Parallel Optimization of Multi-Agent Systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[11] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel Algorithms for Big Data Optimization," Dec. 2013, submitted to *IEEE Transactions on Signal Processing*. [Online]. Available: http://arxiv.org/abs/1402.5521

[12] Z. Quan, S. Cui, H. Poor, and A. Sayed, "Collaborative wideband sensing for cognitive radios," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 60–73, Nov. 2008.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

[14] Y. Yang, M. Zhang, M. Pesavento, and D. P. Palomar, "An online parallel and distributed algorithm for recursive estimation of sparse signals," unpublished.

[15] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[16] "The MOSEK Optimization toolbox for MATLAB manual. Version 7.0." [Online]. Available: http://www.mosek.com/

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.