A Parallel Best-Response Algorithm with Exact Line Search for Nonconvex Sparsity-Regularized Rank Minimization

Yang Yang[1] and Marius Pesavento[2]
1. Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, L-1855 Luxembourg.
2. Communication Systems Group, Technische Universität Darmstadt, Darmstadt 64283, Germany.
Email: yang.yang@uni.lu, pesavento@nt.tu-darmstadt.de

***Abstract*—In this paper, we propose a convergent parallel best-response algorithm with the exact line search for the nondifferentiable nonconvex sparsity-regularized rank minimization problem. On the one hand, it exhibits a faster convergence than subgradient algorithms and block coordinate descent algorithms. On the other hand, its convergence to a stationary point is guaranteed, while ADMM algorithms only converge for convex problems. Furthermore, the exact line search procedure in the proposed algorithm is performed efficiently in closed-form to avoid the meticulous choice of stepsizes, which is however a common bottleneck in subgradient algorithms and successive convex approximation algorithms. Finally, the proposed algorithm is numerically tested.**

***Index Terms*—Big Data Analytics, Line Search, Rank Minimization, Successive Convex Approximation**

I. Introduction

In this paper, we consider the estimation of a low rank matrix $\mathbf{X} \in \mathbb{R}^{N\times K}$ and a sparse matrix $\mathbf{S} \in \mathbb{R}^{I\times K}$ from noisy measurements $\mathbf{Y} \in \mathbb{R}^{N\times K}$ such that

$$\mathbf{Y} = \mathbf{X} + \mathbf{D}\mathbf{S} + \mathbf{V},$$

where $\mathbf{D} \in \mathbb{R}^{N\times I}$ is a known matrix. The rank of $\mathbf{X}$ is much smaller than N and K, i.e, $\text{rank}(\mathbf{X}) \ll \min(N, K)$, and the support size of $\mathbf{S}$ is much smaller than IK, i.e., $\|\mathbf{S}\|_0 \ll IK$.

A natural measure for the data mismatch is the least square error augmented by regularization functions to promote the rank sparsity of $\mathbf{X}$ and support sparsity of $\mathbf{S}$:

$$(\text{SRRM}): \underset{\mathbf{X},\mathbf{S}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{X} + \mathbf{D}\mathbf{S} - \mathbf{Y}\|_F^2 + \lambda\|\mathbf{X}\|_* + \mu\|\mathbf{S}\|_1,$$

where $\|\mathbf{X}\|_*$ is the nuclear norm of $\mathbf{X}$. This sparsity-regularized rank minimization (SRRM) problem plays a fundamental role in the analysis of traffic anomalies in large-scale backbone networks [1]. In this application, $\mathbf{X} = \mathbf{R}\mathbf{Z}$ where $\mathbf{Z}$ is the unknown traffic flows over the time horizon of interest, $\mathbf{R}$ is a given fat routing matrix, $\mathbf{S}$ is the traffic volume anomalies. The matrix $\mathbf{X}$ inherits the rank sparsity from $\mathbf{Z}$ because common temporal patterns among the traffic flows in addition to their periodic behavior render most rows/columns of $\mathbf{Z}$ linearly dependent and thus low rank, and $\mathbf{S}$ is assumed to be sparse because traffic anomalies are expected to happen sporadically and last shortly relative to the measurement interval, which is represented by the number of columns K.

Although problem (SRRM) is convex, it cannot be easily solved by standard solvers when the problem dimension is large, for the reason that the nuclear norm $\|\mathbf{X}\|_*$ is neither differentiable nor decomposable among the blocks of $\mathbf{X}$. It follows from the fact [2, 3]

$$\|\mathbf{X}\|_* = \min_{(\mathbf{P},\mathbf{Q})} \frac{1}{2}\left(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2\right), \text{ s.t. } \mathbf{P}\mathbf{Q} = \mathbf{X}$$

The work of Y. Yang is supported by the ERC project AGNOSTIC, and the work of M. Pesavento is supported by the EXPRESS Project within the DFG Priority Program CoSIP (DFG-SPP 1798).

that it may be useful to consider the following optimization problem where the nuclear norm $\|\mathbf{X}\|_*$ is replaced by $\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2$:

$$\underset{\mathbf{P},\mathbf{Q},\mathbf{S}}{\text{minimize}} \frac{1}{2}\|\mathbf{P}\mathbf{Q} + \mathbf{D}\mathbf{S} - \mathbf{Y}\|_F^2 + \frac{\lambda}{2}\left(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2\right) + \mu\|\mathbf{S}\|_1, \tag{1}$$

where $\mathbf{P} \in \mathbb{R}^{N\times\rho}$ and $\mathbf{Q} \in \mathbb{R}^{\rho\times K}$ for a ρ that is usually much smaller than N and K: $\rho \ll \min(N, K)$. Although problem (1) is nonconvex, it is shown in [4, Prop. 1] that every stationary point of (1) is a global optimal solution of (SRRM) under some mild conditions.

A block coordinate descent (BCD) algorithm is adopted in [5] to find a stationary point of the nonconvex problem (1), where the variables are updated sequentially according to their best-response. For example, when $\mathbf{P}$ (or $\mathbf{Q}$) is updated, the variables $(\mathbf{Q}, \mathbf{S})$ (or $(\mathbf{P}, \mathbf{S})$) are fixed. When $(\mathbf{P}, \mathbf{Q})$ is fixed, the optimization problem w.r.t. $\mathbf{S}$ decouples among its columns:

$$\begin{aligned}&\frac{1}{2}\|\mathbf{P}\mathbf{Q} + \mathbf{D}\mathbf{S} - \mathbf{Y}\|_F^2 + \mu\|\mathbf{S}\|_1 \\ &= \sum_{k=1}^{K}\left(\frac{1}{2}\|\mathbf{P}\mathbf{q}_k - \mathbf{D}\mathbf{s}_k - \mathbf{y}_k\|_2^2 + \mu\|\mathbf{s}_k\|_1\right),\end{aligned}$$

where $\mathbf{q}_k$, $\mathbf{s}_k$ and $\mathbf{y}_k$ is the k-th column of $\mathbf{Q}$, $\mathbf{S}$ and $\mathbf{Y}$, respectively. However, the optimization problem w.r.t. $\mathbf{s}_k$ does not have a closed-form solution and is not easy to solve. To reduce the complexity, the elements of $\mathbf{S}$ are updated row-wise, as the optimization problem w.r.t. $s_{i,k}$, the (i,k)-th element of $\mathbf{S}$, has a closed-form solution:

$$\underset{(s_{i,k})_{k=1}^K}{\text{minimize}} \sum_{k=1}^{K}\left(\frac{1}{2}\|\mathbf{P}\mathbf{q}_k - \mathbf{D}\mathbf{s}_k - \mathbf{y}_k\|_2^2 + \mu\|\mathbf{s}_k\|_1\right).$$

Nevertheless, a drawback of the sequential row-wise update is that it may incur a large delay because the $(i+1)-$th row of $\mathbf{S}$ cannot be updated until the i-th row is updated and the delay may be very large when I is large, which is a norm rather than an exception in big data analytics [6].

The alternating direction method of multipliers (ADMM) algorithm enables the simultaneous update of all elements of $\mathbf{S}$, but it does not have a guarantee convergence to a stationary point because the optimization problem (1) is nonconvex [4]. Note that there is some recent development in ADMM for nonconvex problems, see [7, 8] for example and the references therein. The ADMM algorithm proposed in [7] is designed for nonconvex sharing/consensus problems, and cannot be applied to solve problem (1). The ADMM algorithm proposed in [8] converges if the matrix $\mathbf{D}$ in (1) has full row rank, which is however not necessarily the case.

The nondifferentiable nonconvex problem (1) can also be solved by standard subgradient and/or successive convex approximation (SCA) algorithms [9]. However, convergence of subgradient and SCA algorithms is mostly established under diminishing stepsizes, which is sometimes difficult to deploy in practice because the convergence behavior is sensitive to the decay rate. As a matter of fact, their

applicability in nonsmooth optimization and big data analytics is severely limited by the meticulous choice of stepsizes [6].

In this paper, we propose a convergent parallel best-response algorithm, where all elements of $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{S}$ are updated simultaneously. This is a well known concept in optimization and sometimes listed under different names, for example, the parallel block coordinate descent algorithm (cf. [10]) and the Jacobi algorithm (cf. [11]). To accelerate the convergence, we compute the stepsize by the exact line search procedure proposed in [12]: the exact line search is performed over a properly designed differentiable function and the resulting stepsize can be expressed in a closed-form expression, so that the computational complexity is much lower than the traditional line search which is over the original nondifferentiable objective function. The proposed algorithm has several attractive features: i) the variables are updated simultaneously based on the best-response; ii) the stepsize is computed in closed-form based on the exact line search; iii) it converges to a stationary point, and its advantages over existing algorithms are summarized as follows:

- Feature i) is an advantage over the BCD algorithm;
- Features i) and ii) are advantages over subgradient algorithms;
- Feature ii) is an advantage over SCA algorithms;
- Feature iii) is an advantage over ADMM algorithms.

The above advantages will further be illustrated by numerical results.

II. The Proposed Parallel Best-Response Algorithm with Exact Line Search

In this section, we propose an iterative algorithm to find a stationary point of problem (1). It consists of solving a sequence of successively refined approximate problems, which are presumably much easier to solve than the original problem. To this end, we define

$$f(\mathbf{P},\mathbf{Q},\mathbf{S}) \triangleq \frac{1}{2}\left\|\mathbf{PQ}+\mathbf{DS}-\mathbf{Y}\right\|_F^2+\frac{\lambda}{2}\left(\|\mathbf{P}\|_F^2+\|\mathbf{Q}\|_F^2\right),$$
$$g(\mathbf{S}) \triangleq \mu\left\|\mathbf{S}\right\|_1.$$

Although $f(\mathbf{P},\mathbf{Q},\mathbf{S})$ in (1) is not jointly convex w.r.t. $(\mathbf{P},\mathbf{Q},\mathbf{S})$, it is individual convex in $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{S}$. In other words, $f(\mathbf{P},\mathbf{Q},\mathbf{S})$ is convex w.r.t. one variable while the other two variables are fixed. Preserving and exploiting this partial convexity considerably accelerates the convergence and it has become the central idea in the successive convex approximation and the successive pseudoconvex approximation [11, 12].

To simplify the notation, we use $\mathbf{Z}$ as a compact notation for $(\mathbf{P},\mathbf{Q},\mathbf{S})$: $\mathbf{Z}\triangleq(\mathbf{P},\mathbf{Q},\mathbf{S})$; in the rest of the paper, $\mathbf{Z}$ and $(\mathbf{P},\mathbf{Q},\mathbf{S})$ are used interchangeably. Given $\mathbf{Z}^t=(\mathbf{P}^t,\mathbf{Q}^t,\mathbf{S}^t)$ in iteration t, we approximate the original nonconvex function $f(\mathbf{Z})$ by a convex function $\tilde{f}(\mathbf{Z};\mathbf{Z}^t)$ that is of the following form:

$$\tilde{f}(\mathbf{Z};\mathbf{Z}^t)=\tilde{f}_P(\mathbf{P};\mathbf{Z}^t)+\tilde{f}_Q(\mathbf{Q};\mathbf{Z}^t)+\tilde{f}_S(\mathbf{S};\mathbf{Z}^t), \tag{2}$$

where

$$\tilde{f}_P(\mathbf{P};\mathbf{Z}^t)\triangleq f(\mathbf{P},\mathbf{Q}^t,\mathbf{S}^t)=\frac{1}{2}\left\|\mathbf{PQ}^t+\mathbf{DS}^t-\mathbf{Y}\right\|_F^2+\frac{\lambda}{2}\|\mathbf{P}\|_F^2, \tag{3a}$$

$$\tilde{f}_Q(\mathbf{Q};\mathbf{Z}^t)\triangleq f(\mathbf{P}^t,\mathbf{Q},\mathbf{S}^t)=\frac{1}{2}\left\|\mathbf{P}^t\mathbf{Q}+\mathbf{DS}^t-\mathbf{Y}\right\|_F^2+\frac{\lambda}{2}\|\mathbf{Q}\|_F^2, \tag{3b}$$

$$\begin{aligned}\tilde{f}_S(\mathbf{S};\mathbf{Z}^t)&\triangleq\sum_{i,k}f(\mathbf{P}^t,\mathbf{Q}^t,s_{i,k},(s_{j,k}^t)_{j\neq i},(\mathbf{s}_j^t)_{j\neq i})\\&=\sum_{i,k}\frac{1}{2}\left\|\mathbf{P}^t\mathbf{q}_k^t+\mathbf{d}_i s_{i,k}+\textstyle\sum_{j\neq i}\mathbf{d}_j s_{j,k}^t-\mathbf{y}_k\right\|_2^2\\&=\mathrm{tr}(\mathbf{S}^T\mathbf{d}(\mathbf{D}^T\mathbf{D})\mathbf{S})\\&\quad-\mathrm{tr}(\mathbf{S}^T(\mathbf{d}(\mathbf{D}^T\mathbf{D})\mathbf{S}^t-\mathbf{D}^T(\mathbf{DS}^t-\mathbf{Y}+\mathbf{P}^t\mathbf{Q}^t))),\end{aligned} \tag{3c}$$

with $\mathbf{q}_k$ (or $\mathbf{y}_k$) and $\mathbf{d}_i$ denoting the k-th and i-th column of $\mathbf{Q}$ (or $\mathbf{Y}$) and $\mathbf{D}$, respectively, while $\mathbf{d}(\mathbf{D}^T\mathbf{D})$ denotes a diagonal matrix with elements on the main diagonal identical to those of the matrix $\mathbf{D}^T\mathbf{D}$. Note that in the approximate function w.r.t. $\mathbf{P}$ and $\mathbf{Q}$, the remaining variables $(\mathbf{Q},\mathbf{S})$ and $(\mathbf{P},\mathbf{S})$ are fixed, respectively. Although it is tempting to define the approximate function of $f(\mathbf{P},\mathbf{Q},\mathbf{S})$ w.r.t. $\mathbf{S}$ by fixing $\mathbf{P}$ and $\mathbf{Q}$, minimizing $f(\mathbf{P}^t,\mathbf{Q}^t,\mathbf{S})$ w.r.t. the matrix variable $\mathbf{S}$ does not have a closed-form solution and must be solved iteratively. Therefore the proposed approximate function $\tilde{f}_S(\mathbf{S};\mathbf{Z}^t)$ in (3c) consists of IK component functions, and in the (i,k)-th component function, $s_{i,k}$ is the variable while all other variables are fixed, namely, $\mathbf{P}$, $\mathbf{Q}$, $(s_{j,k})_{j\neq i}$, and $(\mathbf{s}_j)_{j\neq i}$. As we will show shortly, minimizing $\tilde{f}(\mathbf{S};\mathbf{Z}^t)$ w.r.t. $\mathbf{S}$ exhibits a closed-form solution.

We remark that the approximate function $\tilde{f}(\mathbf{Z};\mathbf{Z}^t)$ is a (strongly) convex function and it is differentiable in both $\mathbf{Z}$ and $\mathbf{Z}^t$. Furthermore, the gradient of the approximate function $\tilde{f}(\mathbf{P},\mathbf{Q},\mathbf{S};\mathbf{Z}^t)$ is equal to that of $f(\mathbf{P},\mathbf{Q},\mathbf{S})$ at $\mathbf{Z}=\mathbf{Z}^t$. To see this:

$$\nabla_{\mathbf{P}}\tilde{f}(\mathbf{Z};\mathbf{Z}^t)=\nabla_{\mathbf{P}}\tilde{f}_P(\mathbf{P};\mathbf{Z}^t)=\nabla_{\mathbf{P}}\ f(\mathbf{P},\mathbf{Q}^t,\mathbf{S}^t)\big|_{\mathbf{P}=\mathbf{P}^t},$$

and similarly $\nabla_{\mathbf{Q}}\tilde{f}(\mathbf{Z};\mathbf{Z}^t)=\nabla_{\mathbf{Q}}f(\mathbf{P},\mathbf{Q},\mathbf{S})|_{\mathbf{Z}=\mathbf{Z}^t}$. Furthermore, $\nabla_{\mathbf{S}}\tilde{f}(\mathbf{Z};\mathbf{Z}^t)=(\nabla_{s_{i,k}}\tilde{f}(\mathbf{Z};\mathbf{Z}^t))_{i,k}$ while

$$\begin{aligned}\nabla_{s_{i,k}}\tilde{f}(\mathbf{Z};\mathbf{Z}^t)&=\nabla_{s_{i,k}}\tilde{f}_S(\mathbf{S};\mathbf{Z}^t)\\&=\nabla_{s_{i,k}}f(\mathbf{P}^t,\mathbf{Q}^t,s_{i,k},\mathbf{s}_{i,-k}^t,\mathbf{s}_{-i}^t)\\&=\nabla_{s_{i,k}}f(\mathbf{P},\mathbf{Q},\mathbf{S})\big|_{\mathbf{Z}=\mathbf{Z}^t}.\end{aligned}$$

In iteration t, the approximate problem consists of minimizing the approximate function over the same feasible set as the original problem (1):

$$\underset{\mathbf{Z}=(\mathbf{P},\mathbf{Q},\mathbf{S})}{\text{minimize}}\quad\tilde{f}(\mathbf{Z};\mathbf{Z}^t)+g(\mathbf{S}). \tag{4}$$

Since $\tilde{f}(\mathbf{Z};\mathbf{Z}^t)$ is strongly convex in $\mathbf{Z}$ and $g(\mathbf{S})$ is a convex function w.r.t. $\mathbf{S}$, the approximate problem (4) is convex and it has a unique (globally) optimal solution, which is denoted as $\mathbb{B}\mathbf{Z}^t=(\mathbb{B}_P\mathbf{Z}^t,\mathbb{B}_Q\mathbf{Z}^t,\mathbb{B}_S\mathbf{Z}^t)$.

The approximate problem (4) naturally decomposes into several smaller problems which can be solved in parallel:

$$\begin{aligned}\mathbb{B}_P\mathbf{Z}^t&\triangleq\arg\min_{\mathbf{P}_k}\tilde{f}_P(\mathbf{P};\mathbf{Z}^t)\\&=(\mathbf{Y}-\mathbf{DS}^t)(\mathbf{Q}^t)^T(\mathbf{Q}^t(\mathbf{Q}^t)^T+\lambda\mathbf{I})^{-1},\end{aligned} \tag{5a}$$

$$\begin{aligned}\mathbb{B}_Q\mathbf{Z}^t&\triangleq\arg\min_{\mathbf{Q}}\tilde{f}_Q(\mathbf{Q};\mathbf{Z}^t)\\&=((\mathbf{P}^t)^T\mathbf{P}^t+\lambda\mathbf{I})^{-1}(\mathbf{P}^t)^T(\mathbf{Y}-\mathbf{DS}^t),\end{aligned} \tag{5b}$$

$$\begin{aligned}\mathbb{B}_S\mathbf{Z}^t&\triangleq\arg\min_{\mathbf{S}}\tilde{f}_S(\mathbf{S};\mathbf{Z}^t)+g(\mathbf{S})\\&=\mathbf{d}(\mathbf{D}^T\mathbf{D})^{-1}\cdot\\&\quad\mathcal{S}_\mu\left(\mathbf{d}(\mathbf{D}^T\mathbf{D})\mathbf{S}^t-\mathbf{D}^T(\mathbf{DS}^t-\mathbf{Y}^t+\mathbf{P}^t\mathbf{Q}^t)\right),\end{aligned} \tag{5c}$$

where $\mathcal{S}_\mu(\mathbf{X})$ is an element-wise soft-thresholding operator: the (i,j)-th element of $\mathcal{S}_\mu(\mathbf{X})$ is $[X_{ij}-\lambda]^+-[-X_{ij}-\lambda]^+$. As we can readily see from (5), the approximate problems can be solved efficiently because the optimal solutions are provided in an analytical expression.

Since $\tilde{f}(\mathbf{Z};\mathbf{Z}^t)$ is convex in $\mathbf{Z}$ and differentiable in both $\mathbf{Z}$ and $\mathbf{Z}^t$, and has the same gradient as $f(\mathbf{Z})$ at $\mathbf{Z}=\mathbf{Z}^t$, it follows from [12, Prop. 1] that $\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t$ is a descent direction of the original

objective function $f(\mathbf{Z})+g(\mathbf{S})$ at $\mathbf{Z}=\mathbf{Z}^t$. The variable update in the t-th iteration is thus defined as follows:

$$\mathbf{P}^{t+1}=\mathbf{P}^t+\gamma(\mathbb{B}_P\mathbf{Z}^t-\mathbf{P}^t), \tag{6a}$$
$$\mathbf{Q}^{t+1}=\mathbf{Q}^t+\gamma(\mathbb{B}_Q\mathbf{Z}^t-\mathbf{Q}^t), \tag{6b}$$
$$\mathbf{S}^{t+1}=\mathbf{S}^t+\gamma(\mathbb{B}_S\mathbf{Z}^t-\mathbf{S}^t), \tag{6c}$$

where $\gamma\in(0,1]$ is the stepsize that should be properly selected.

A natural (and traditional) choice of the stepsize γ is given by the exact line search:

$$\min_{0\le\gamma\le1}\left\{f(\mathbf{Z}^t+\gamma(\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t))+g(\mathbf{S}^t+\gamma(\mathbb{B}_S\mathbf{Z}^t-\mathbf{S}^t))\right\}, \tag{7}$$

in which the stepsize that yields the largest decrease in objective function value along the direction $\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t$ is selected. Nevertheless, this choice leads to high computational complexity, because $g(\mathbf{S})$ is nondifferentiable and the exact line search involves minimizing a nondifferentiable function. Alternatives include constant stepsizes and diminishing stepsizes. However, they suffer from slow convergence (cf. [11]) and parameter tuning (cf. [12]). As a matter of fact, the meticulous choice of stepsizes have become a major bottleneck for subgradient and successive convex approximation algorithm [6].

It is shown in [12, Sec. III-A] that to achieve convergence, it suffices to perform the exact line search over the following differentiable function:

$$f(\mathbf{Z}^t+\gamma(\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t))+g(\mathbf{S}^t)+\gamma(g(\mathbb{B}_S\mathbf{Z}^t)-g(\mathbf{S}^t)), \tag{8}$$

which is an upper bound of the objective function in (7) after applying Jensen's inequality to the convex nondifferentiable function $g(\mathbf{S})$:

$$g(\mathbf{S}^t+\gamma(\mathbb{B}_S\mathbf{Z}^t-\mathbf{S}^t))\le g(\mathbf{S}^t)+\gamma(g(\mathbb{B}_S\mathbf{Z}^t)-g(\mathbf{S}^t)).$$

This exact line search procedure over the differentiable function (8) achieves a good tradeoff between performance and complexity. Furthermore, after substituting the expressions of $f(\mathbf{Z})$ and $g(\mathbf{S})$ into (8), the exact line search boils down to minimizing a four order polynomial over the interval $[0,1]$:

$$\begin{aligned}\gamma^t&=\arg\min_{0\le\gamma\le1}\left\{f(\mathbf{Z}^t+\gamma(\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t))+\gamma(g(\mathbb{B}_S\mathbf{X}^t)-g(\mathbf{S}^t))\right\}\\&=\arg\min_{0\le\gamma\le1}\left\{\frac{1}{4}a\gamma^4+\frac{1}{3}b\gamma^3+\frac{1}{2}c\gamma^2+d\gamma\right\},\end{aligned} \tag{9}$$

where

$$\begin{aligned}a&\triangleq 2\left\|\triangle\mathbf{P}^t\triangle\mathbf{Q}^t\right\|_F^2,\\b&\triangleq 3\mathrm{tr}(\triangle\mathbf{P}^t\triangle\mathbf{Q}^t(\mathbf{P}^t\triangle\mathbf{Q}^t+\triangle\mathbf{P}^t\mathbf{Q}^t+\mathbf{D}\triangle\mathbf{S}^t)^T),\\c&\triangleq 2\mathrm{tr}(\triangle\mathbf{P}^t\triangle\mathbf{Q}^t(\mathbf{P}^t\mathbf{Q}^t+\mathbf{D}\mathbf{S}^t-\mathbf{Y}^t)^T)\\&\quad+\left\|\mathbf{P}^t\triangle\mathbf{Q}^t+\triangle\mathbf{P}^t\mathbf{Q}^t+\mathbf{D}\triangle\mathbf{S}^t\right\|_F^2\\&\quad+\lambda(\left\|\triangle\mathbf{P}^t\right\|_F^2+\left\|\triangle\mathbf{Q}^t\right\|_F^2),\\d&\triangleq\mathrm{tr}((\mathbf{P}^t\triangle\mathbf{Q}^t+\triangle\mathbf{P}^t\mathbf{Q}^t+\mathbf{D}\triangle\mathbf{S}^t)(\mathbf{P}^t\mathbf{Q}^t+\mathbf{D}\mathbf{S}^t-\mathbf{Y}^t))\\&\quad+\lambda(\mathrm{tr}(\mathbf{P}^t\triangle\mathbf{P}^t)+\mathrm{tr}(\mathbf{Q}^t\triangle\mathbf{Q}^t))+\mu(\left\|\mathbb{B}_S\mathbf{X}^t\right\|_1-\left\|\mathbf{S}^t\right\|_1),\end{aligned}$$

for $\triangle\mathbf{P}^t\triangleq\mathbb{B}_P\mathbf{Z}^t-\mathbf{P}^t$, $\triangle\mathbf{Q}^t\triangleq\mathbb{B}_Q\mathbf{Z}^t-\mathbf{Q}^t$ and $\triangle\mathbf{S}^t\triangleq\mathbb{B}_S\mathbf{Z}^t-\mathbf{S}^t$. Finding the optimal points of (9) is equivalent to finding the nonnegative real root of a third-order polynomial. Making use of Cardano's method, we could express γ^t defined in (9) in a closed-form expression:

$$\gamma^t=[\bar{\gamma}^t]_0^1, \tag{10a}$$
$$\bar{\gamma}^t=\sqrt[3]{\Sigma_1+\sqrt{\Sigma_1^2+\Sigma_2^3}}+\sqrt[3]{\Sigma_1-\sqrt{\Sigma_1^2+\Sigma_2^3}}-\frac{b}{3a}, \tag{10b}$$

where $[x]_0^1=\max(\min(x,1),0)$ is the projection of x onto the interval $[0,1]$, $\Sigma_1\triangleq-(b/3a)^3+bc/6a^2-d/2a$ and $\Sigma_2\triangleq c/3a-(b/3a)^2$. Note that in (10b), the right hand side has three values (two of them could be complex numbers), and the equal sign reads to be equal to the smallest one among the real nonnegative values.

Algorithm 1 The parallel best-response algorithm with exact line search for problem (1)

Data: $t=0$, $\mathbf{Z}^0$ (arbitrary but fixed), stop criterion δ.
S1: Compute $(\mathbb{B}_P\mathbf{Z}^t,\mathbb{B}_Q\mathbf{Z}^t,\mathbb{B}_S\mathbf{Z}^t)$ according to (5).
S2: Determine the stepsize γ^t by the exact line search (10).
S3: Update $(\mathbf{P},\mathbf{Q},\mathbf{Z})$ according to (6).
S4: If $\left|\mathrm{tr}((\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t)^T\nabla f(\mathbf{Z}^t))+g(\mathbb{B}_S\mathbf{Z}^t)-g(\mathbf{S}^t)\right|\le\delta$, STOP; otherwise $t\leftarrow t+1$ and go to **S1**.

The proposed algorithm is summarized in Algorithm 1, and we draw a few comments on its features and advantages.

On the parallel best-response update: In each iteration, the variables $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{S}$ are updated simultaneously based on the best-response. The improvement in convergence speed w.r.t. the BCD algorithm in [5] is notable because in the BCD algorithm, the optimization w.r.t. each element of $\mathbf{S}$, say $s_{i,k}$, is implemented in a sequential order, and the number of elements, IK, is usually very large in big data applications. To avoid the meticulous choice of stepsizes and further accelerate the convergence, the exact line search is performed over the differentiable function $f(\mathbf{Z}^t+\gamma(\mathbb{B}\mathbf{Z}^t-\mathbf{Z}^t))+\gamma(g(\mathbb{B}_S\mathbf{Z}^t)-g_S(\mathbf{Z}^t))$ and it can be computed by a closed-form expression. The yields easier implementation and faster convergence than subgradient and SCA algorithms with diminishing stepsizes.

On the complexity: The complexity of the proposed algorithm is maintained at a very low level, because both the best-responses $(\mathbb{B}_P\mathbf{Z}^t,\mathbb{B}_Q\mathbf{Z}^t,\mathbb{B}_S\mathbf{Z}^t)$ and the exact line search can be computed by closed-form expressions, cf. (3) and (10). Note that computing $\mathbb{B}_P\mathbf{Z}^t$ and $\mathbb{B}_Q\mathbf{Z}^t$ according to (5a)-(5b) involves a matrix inverse. This is usually affordable because the matrices to be inverted are of a dimension $\rho\times\rho$ while the rank ρ is usually small. Furthermore, the matrix inverse operation could be saved by adopting an element-wise decomposition for $\mathbf{P}$ and $\mathbf{Q}$ that is in the same essence as $\mathbf{S}$ in (3c).

On the convergence: The proposed Algorithm 1 has a guaranteed convergence in the sense that every limit point of the sequence $\{\mathbf{Z}^t\}_t$ is a stationary point of problem (1). This claim directly follows from [12, Theorem 1], and it serves as a certificate for the solution quality.

A. Decomposition of the Proposed Algorithm

The proposed Algorithm 1 can be further decomposed to enable the parallel processing over a number of L nodes in a distributed network. To see this, we first decompose the matrix variables $\mathbf{P}$, $\mathbf{D}$ and $\mathbf{Y}$ into multiple blocks $(\mathbf{P}_l)_{l=1}^L$, $(\mathbf{D}_l)_{l=1}^L$ and $(\mathbf{Y}_l)_{l=1}^L$, while $\mathbf{P}_l\in\mathbb{R}^{N_l\times\rho}$, $\mathbf{D}_l\in\mathbb{R}^{N_l\times I}$ and $\mathbf{Y}_l\in\mathbb{R}^{N_l\times K}$ consists of N_l rows of $\mathbf{P}$, $\mathbf{D}$ and $\mathbf{Y}$, respectively:

$$\mathbf{P}=\begin{bmatrix}\mathbf{P}_1\\\mathbf{P}_2\\\vdots\\\mathbf{P}_L\end{bmatrix},\mathbf{D}=\begin{bmatrix}\mathbf{D}_1\\\mathbf{D}_2\\\vdots\\\mathbf{D}_L\end{bmatrix},\mathbf{Y}=\begin{bmatrix}\mathbf{Y}_1\\\mathbf{Y}_2\\\vdots\\\mathbf{Y}_L\end{bmatrix},$$

where each node l has access to the variables $(\mathbf{P}_l,\mathbf{Q},\mathbf{S})$. The computation of $\mathbb{B}_P\mathbf{Z}^t$ in (6a) can be decomposed as $\mathbb{B}_P\mathbf{Z}^t=(\mathbb{B}_{P,l}\mathbf{Z}^t)_{l=1}^L$:

$$\mathbb{B}_{P,l}\mathbf{Z}^t=(\mathbf{Y}_l-\mathbf{D}_l\mathbf{S}^t)(\mathbf{Q}^t)^T(\mathbf{Q}^t(\mathbf{Q}^t)^T+\lambda\mathbf{I})^{-1},l=1,\ldots,L.$$

Accordingly, the computation of $\mathbb{B}_Q\mathbf{Z}^t$ and $\mathbb{B}_S\mathbf{Z}^t$ in (6b) and (6c) can be rewritten as

$$\mathbb{B}_Q\mathbf{Z}^t=\left(\sum_{l=1}^L(\mathbf{P}_l^t)^T\mathbf{P}_l^t+\lambda\mathbf{I}\right)^{-1}\left(\sum_{l=1}^L(\mathbf{P}_l^t)^T(\mathbf{Y}_l-\mathbf{D}_l\mathbf{S}^t)\right),$$

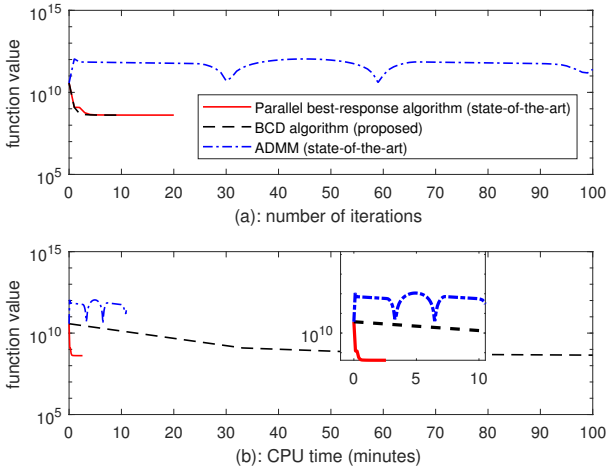

Figure 1. objective function value versus the number of iterations and CPU time in minutes.

$$\mathbb{B}_S\mathbf{Z}^t = \mathbf{d}\left(\sum_{l=1}^L \mathbf{D}_l^T\mathbf{D}_l\right)^{-1}\cdot \mathcal{S}_\mu\left(\mathbf{d}\left(\sum_{l=1}^L \mathbf{D}_l^T\mathbf{D}_l\right)\mathbf{S}^t - \sum_{l=1}^L \mathbf{D}_l^T(\mathbf{D}_l\mathbf{S}^t - \mathbf{Y}_l^t + \mathbf{P}_l^t\mathbf{Q}^t)\right).$$

Before determining the stepsize, the computation of a in (10) can also be decomposed among the nodes as $a = \sum_{l=1}^L a_l$, where

$$a_l \triangleq 2\left\|\triangle\mathbf{P}_l^t\triangle\mathbf{Q}^t\right\|_F^2.$$

The decomposition of b, c, and d is similar to that of a, where

$$\begin{aligned} b_l &\triangleq 3\text{tr}(\triangle\mathbf{P}_l^t\triangle\mathbf{Q}^t(\mathbf{P}_l^t\triangle\mathbf{Q}^t + \triangle\mathbf{P}_l^t\mathbf{Q}^t + \mathbf{D}_l\triangle\mathbf{S}^t)^T),\\ c_l &\triangleq 2\text{tr}(\triangle\mathbf{P}_l^t\triangle\mathbf{Q}^t(\mathbf{P}_l^t\mathbf{Q}^t + \mathbf{D}_l\mathbf{S}^t - \mathbf{Y}_l^t)^T)\\ &\quad + \left\|\mathbf{P}_l^t\triangle\mathbf{Q}^t + \triangle\mathbf{P}_l^t\mathbf{Q}^t + \mathbf{D}_l\triangle\mathbf{S}^t\right\|_F^2\\ &\quad + \lambda\left\|\triangle\mathbf{P}_l^t\right\|_F^2 + \frac{\lambda}{I}\left\|\triangle\mathbf{Q}_l^t\right\|_F^2,\\ d_l &\triangleq \text{tr}((\mathbf{P}_l^t\triangle\mathbf{Q}^t + \triangle\mathbf{P}_l^t\mathbf{Q}^t + \mathbf{D}_l\triangle\mathbf{S}^t)(\mathbf{P}_l^t\mathbf{Q}^t + \mathbf{D}_l\mathbf{S}^t - \mathbf{Y}_l^t))\\ &\quad + \lambda\text{tr}(\mathbf{P}_l^t\triangle\mathbf{P}_l^t) + \frac{\lambda}{I}\text{tr}(\mathbf{Q}^t\triangle\mathbf{Q}^t) + \frac{\mu}{I}(\left\|\mathbb{B}_S\mathbf{X}^t\right\|_1 - \left\|\mathbf{S}^t\right\|_1). \end{aligned}$$

To compute the stepsize as in (10), the nodes mutually exchange (a_l, b_l, c_l, d_l). The four dimensional vector (a_l, b_l, c_l, d_l) provides each node with all the necessary information to individually calculate (a, b, c, d) and $(\Sigma_1, \Sigma_2, \Sigma_3)$, and then the stepsize γ^t according to (10). The signaling incurred by the exact line search is thus small and affordable.

III. Numerical Simulations

In this section, we perform numerical tests to compare the proposed Algorithm 1 with the BCD algorithm [5] and the ADMM algorithm [4]. We start with a brief description of the ADMM algorithm: the problem (1) can be rewritten as

$$\begin{aligned} \underset{\mathbf{P},\mathbf{Q},\mathbf{A},\mathbf{B}}{\text{minimize}} \quad & \frac{1}{2}\left\|\mathbf{PQ} + \mathbf{DA} - \mathbf{Y}\right\|_F^2 + \frac{\lambda}{2}\left(\left\|\mathbf{P}\right\|_F^2 + \left\|\mathbf{Q}\right\|_F^2\right) + \mu\left\|\mathbf{B}\right\|_1\\ \text{subject to} \quad & \mathbf{A} = \mathbf{B}. \end{aligned} \tag{11}$$

The augmented Lagrangian of (11) is

$$\begin{aligned} L_c(\mathbf{P},\mathbf{Q},\mathbf{A},\mathbf{B},\mathbf{\Pi}) =& \frac{1}{2}\left\|\mathbf{PQ} + \mathbf{DA} - \mathbf{Y}\right\|_F^2 + \frac{\lambda}{2}\left(\left\|\mathbf{P}\right\|_F^2 + \left\|\mathbf{Q}\right\|_F^2\right)\\ &+ \mu\left\|\mathbf{B}\right\|_1 + \text{tr}(\mathbf{\Pi}^T(\mathbf{A} - \mathbf{B})) + \frac{c}{2}\left\|\mathbf{A} - \mathbf{B}\right\|_F^2, \end{aligned}$$

The code is available at http://orbilu.uni.lu/handle/10993/33772.

where c is a positive constant. In ADMM, the variables are updated in the t-th iteration as follows:

$$\begin{aligned} (\mathbf{Q}^{t+1}, \mathbf{B}^{t+1}) &= \arg\min_{\mathbf{Q},\mathbf{A}} L_c(\mathbf{P}^t, \mathbf{Q}, \mathbf{A}^t, \mathbf{B}, \mathbf{\Pi}^t),\\ \mathbf{P}^{t+1} &= \arg\min_{\mathbf{P}} L_c(\mathbf{P}, \mathbf{Q}^{t+1}, \mathbf{A}^{t+1}, \mathbf{B}^t, \mathbf{\Pi}^t),\\ \mathbf{A}^{t+1} &= \arg\min_{\mathbf{B}} L_c(\mathbf{P}^{t+1}, \mathbf{Q}^{t+1}, \mathbf{A}, \mathbf{B}^{t+1}, \mathbf{\Pi}^t),\\ \mathbf{\Pi}^{t+1} &= \mathbf{\Pi}^t + c(\mathbf{A}^{t+1} - \mathbf{B}^{t+1}). \end{aligned}$$

Note that the solutions to the above optimization problems have an analytical expression [4]. We set $c = 10^4$.

The simulation parameters are set as follows. $N = 1000$, $K = 4000$, $I = 4000$, $\rho = 10$. The elements of $\mathbf{D}$ are generated randomly and they are either 0 or 1. The elements of $\mathbf{V}$ follow the Gaussian distribution with mean 0 and variance 0.01. Each element of $\mathbf{S}$ can take three possible values, namely, -1, 0,1, with the probability $P(S_{i,k} = -1) = P(S_{ik} = 1) = 0.05$ and $P(S_{ik} = 0) = 0.9$. We set $\mathbf{Y} = \mathbf{PQ} + \mathbf{DS} + \mathbf{V}$, where $\mathbf{P}$ and $\mathbf{Q}$ are generated randomly following the Gaussian distribution $\mathcal{N}(\mathbf{0}, 100/I)$ and $\mathcal{N}(\mathbf{0}, 100/K)$, respectively. The sparsity regularization parameters are $\lambda = 0.1\cdot\|\mathbf{Y}\|$ ($\|\mathbf{Y}\|$ is the spectral norm of $\mathbf{Y}$) and $\mu = 0.1\cdot\left\|\mathbf{D}^T\mathbf{Y}\right\|_\infty$. The simulation results are averaged over 10 realizations.

In Figure 1 (a), we show the objective function value versus the number of iterations achieved by different algorithms. As we can see from Figure 1, the ADMM does not converge, as the optimization problem (11) (and (1)) is nonconvex. We also observe that the behavior of the ADMM is very sensitive to the value of c: in some instances, the ADMM may converge if c is large enough, but it is a difficult task on its own to choose an appropriate value of c to achieve a good performance.

Note that for the BCD algorithm in Figure 1, all elements of $\mathbf{S}$ are updated once, in a sequential order, in one iteration. We can see from Figure 1 (a) that the BCD algorithm converges in the same number of iterations as the proposed Algorithm 1. But the incurred delay of each iteration in the BCD algorithm is typically very large, because all rows are updated sequentially. On the other hand, in the proposed algorithm, all variables are updated simultaneously and the CPU time (in minutes) needed for each iteration is relatively small. For the visual convenience in Figure 1 (b), the curve of the proposed algorithm is magnified in a small window, and the curve of the BCD algorithm is plotted for the first 100 minutes only. We see from Figure 1 (b) that the proposed algorithm converges to a stationary point in less than 1 minute, while it takes the BCD algorithm about 100 minutes to find a solution that is reasonably good. Furthermore, it takes the BCD algorithm about 5 hours to converge to a solution that is as good as the proposed algorithm. This marks a notable improvement which is important in real time big data applications.

IV. Concluding Remarks

In this paper, we have proposed a parallel best-response algorithm for the nonconvex sparsity-regularized rank minimization problem. The proposed algorithm exhibits fast convergence and low complexity, because 1) the variables are updated simultaneously based on their best response; 2) the stepsize is based on the exact line search and it is performed over a differentiable function; and 3) both the best response and the stepsize are computed by closed-form expressions. Furthermore, the proposed algorithm has a guaranteed convergence to a stationary point. Because of these attractive features, the proposed algorithm are easy to implement and perform well in different settings. Numerical results consolidate the advantages of the proposed algorithm, especially the notable improvement in the CPU time compared with the state-of-the-art BCD algorithm. To promote reproducible research, the simulation code is made available online.

REFERENCES

[1] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5186–5205, 2013.

[2] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, jan 2010.

[3] C. Steffens and M. Pesavento, "Block- and Rank-Sparse Recovery for Direction Finding in Partly Calibrated Arrays," pp. 1–29, 2017.

[4] M. Mardani, G. Mateos, and G. B. Giannakis, "Decentralized sparsity-regularized rank minimization: Algorithms and applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5374–5388, 2013.

[5] ——, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE Journal on Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 50–66, 2013.

[6] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and Optimization for Big Data Analytics: (Statistical) learning tools for our era of data deluge," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 18–31, sep 2014.

[7] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence Analysis of Alternating Direction Method of Multipliers for a Family of Nonconvex Problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, jan 2016.

[8] B. Jiang, T. Lin, S. Ma, and S. Zhang, "Structured Nonconvex and Nonsmooth Optimization: Algorithms and Iteration Complexity Analysis," pp. 1–30, 2016.

[9] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel Selective Algorithms for Nonconvex Big Data Optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1874–1889, nov 2015.

[10] M. Elad, "Why simple shrinkage is still relevant for redundant representations?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5559–5569, 2006.

[11] 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.

[12] Y. Yang and M. Pesavento, "A Unified Successive Pseudoconvex Approximation Framework," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3313–3328, 2017.