A PROXIMAL GRADIENT ALGORITHM FOR DECENTRALIZED NONDIFFERENTIABLE OPTIMIZATION

Wei Shi[*] *Qing Ling*[*] *Gang Wu*[*] *Wotao Yin*[†]

[*] Department of Automation, University of Science and Technology of China, Hefei, Anhui, China
[†] Department of Mathematics, University of California, Los Angeles, California, USA

ABSTRACT

In this paper, we focus on solving the decentralized consensus optimization problem defined over a networked multi-agent system. All the agents shall cooperatively find a common minimizer of the overall objective while each agent holds its own local objective and can only communicate with its neighbors. Motivated by many applications in which the local objective is the sum of a differentiable part and a nondifferentiable part, this paper proposes a proximal gradient exact first-order algorithm (PG-EXTRA) that utilizes the separable problem structure. Here, "exact" means this decentralized algorithm yields an exact consensus minimizer using a fixed step size. When the nondifferentiable part vanishes, PG-EXTRA reduces to EXTRA, an existing decentralized optimization algorithm. When the differentiable part vanishes, PG-EXTRA finds its special case P-EXTRA, a proximal algorithm. We prove convergence and rate of convergence for PG-EXTRA. Numerical experiments on a decentralized compressive sensing problem validates the theoretical results.

Index Terms— Multi-agent network, decentralized optimization, proximal gradient method

1. INTRODUCTION

This paper considers a connected network constituted by n agents that cooperatively solve the *decentralized consensus optimization* problem in the form

$$\underset{x\in\mathbb{R}^p}{\text{minimize}}\ \bar{f}(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x). \tag{1}$$

Here x is a p-dimensional optimization variable common to all the agents, and $f_i:\mathbb{R}^p\to\mathbb{R}$ is a convex function privately known by agent i. In decentralized optimization, each agent uses information from itself and its neighbors (i.e., those agents with whom it has one-hop links) to update its local iterate, and hence avoids resource-demanding multi-hop communication. All the agents are expected to eventually obtain a consensual solution of (1).

We focus on the case that for each agent i, the local objective f_i is the sum of a Lipschitz-differentiable function $s_i:\mathbb{R}^p\to\mathbb{R}$ and a possibly nondifferentiable function $r_i:\mathbb{R}^p\to\mathbb{R}$

$$f_i(x)=s_i(x)+r_i(x). \tag{2}$$

We are particularly interested in r_i's whose proximal point problems have explicit solutions. To be specific, given a point $y\in\mathbb{R}^p$ and a scalar $\alpha>0$, it is assumed easy to solve

$$\underset{x\in\mathbb{R}^p}{\text{minimize}}\ r_i(x)+\frac{1}{2\alpha}\|x-y\|_2^2. \tag{3}$$

Through utilizing the special structures of the objective functions as described in (2) and (3), this paper aims to develop an efficient decentralized proximal gradient algorithm for (1), which enjoys the benefits of fast convergence speed and low computation cost.

1.1. Related Work

Different from centralized processing that requires a fusion center to collect data and make decisions, decentralized approaches rely on information exchange between neighbors in the network and autonomous optimization by individual agents, and are hence robust to failure of critical relaying agents and scalable to large-scale networks. These compelling advantages lead to wide applications of decentralized optimization in robotic networks [1, 2], wireless sensor networks [3, 4], smart grids [5, 6], and distributed machine learning systems [7, 8], to name a few. In these applications, the decentralized consensus optimization problem in the form of (1) arises as a generic model.

Existing algorithms that solve (1) can be classified into two categories. The first category includes those algorithms with implicit updates. At each iteration, each agent solves an optimization subproblem whose objective is the local objective function plus some other term determined by its neighboring iterates; the term is linear in the dual decomposition method [9] or quadratic in the decentralized alternating direction method of multipliers (ADMM) [3, 10]. These algorithms require agents to have sufficient computing powers as implicit updates are generally expensive. The second category includes those algorithms with explicit updates. At each iteration, each agent combines iterates from itself and its neighbors and runs a local (sub)gradient step. This can be done by modifying the classical (sub)gradient or dual averaging algorithms to their decentralized versions [11, 12], or by solving the subproblems of ADMM in an inexact manner [13, 14]. The decentralized gradient method encounters the dilemma of either using diminishing step size for accurate but slow convergence [15], or using a constant step size for fast but inaccurate convergence [16]. To address this issue, [17] proposes an exact first-order algorithm (EXTRA) that introduces computationally inexpensive compensations to the gradient descent steps to cancel network-wide gradient noise, and achieves fast yet exact convergence through using a constant step size, which is independent of the network size.

When the local objective functions have the composite forms as (2), the existing decentralized consensus optimization algorithms are inefficient, with either a high per-iteration cost or a slow convergence speed. The algorithms with implicit updates are unable to utilize the special composite structures and thus need to solve difficult subproblems. While the algorithms with explicit updates have to use subgradients due to the nondifferentiable functions r_i, which often results in slow convergence. Observe that the composite forms

appear in various applications; examples include: 1) in a geometric median problem, s_i is null and r_i is an ℓ_2 norm term [18, 19]; 2) in a compressive sensing problem, s_i is a data fidelity term such as squared ℓ_2 norm, and r_i is a sparsity-promoting regularization term such as ℓ_1 norm [4, 20]; 3) in a constrained optimization problem, s_i is a certain differentiable objective function and r_i is an indicator function whose value is zero when the solution is in the feasible set and infinite otherwise [21, 22, 23].

1.2. Contributions and Paper Organization

To address the issue that each local objective function is the summation of a differentiable part and a nondifferentiable part, this paper develops PG-EXTRA, a proximal gradient version of EXTRA (Section 2). Taking advantages of the separable structures of the local objective function, PG-EXTRA lets each agent combine iterates of itself and its neighbors, run a gradient descent-ascent step with respect to the differentiable function, and then run a proximal step with respect to the nondifferentiable function (Section 2.1). When the nondifferentiable function vanishes, PG-EXTRA reduces to EXTRA; when the differentiable function is vanishes, PG-EXTRA becomes P-EXTRA, a proximal version of EXTRA without the gradient steps (Section 2.2).

Section 3 establishes convergence and rate of convergence for PG-EXTRA and P-EXTRA. When the differentiable parts s_i have Lipschitz continuous gradients, PG-EXTRA converges to the solution set and the rate is $o(1/k)$ where k is the number of iteration. Similar results hold for P-EXTRA but its convergence is stronger than PG-EXTRA in the sense that P-EXTRA allows an arbitrary positive step size. Performance of PG-EXTRA is demonstrated in Section 4. Simulation results on the decentralized compressive sensing problem confirm theoretical findings and validate the effectiveness of PG-EXTRA.

1.3. Notation

Throughout the paper, we let agent i hold a *local copy* $x_{(i)} \in \mathbb{R}^p$ of the global variable x; its value at iteration k is denoted by $x_{(i)}^k$. We introduce an aggregate objective function of the local variables

$$\mathbf{f}(\mathbf{x}) \triangleq \sum_{i=1}^n f_i(x_{(i)}),$$

where $bx \triangleq \left[x_{(1)}^{\mathrm{T}}; \cdots; x_{(n)}^{\mathrm{T}}\right] \in \mathbb{R}^{n\times p}$. Each row i of $\mathbf{x}$ corresponds to agent i. We say that $\mathbf{x}$ is *consensual* if all of its rows are identical, i.e., $x_{(1)} = \cdots = x_{(n)}$. Similar to the definition of $\mathbf{f}(\mathbf{x})$, define the differentiable and nondifferentiable parts of the aggregate objective function as

$$\mathbf{s}(\mathbf{x}) \triangleq \sum_{i=1}^n s_i(x_{(i)}) \quad \text{and} \quad \mathbf{r}(\mathbf{x}) \triangleq \sum_{i=1}^n r_i(x_{(i)}),$$

respectively. By definition $\mathbf{f}(\mathbf{x}) = \mathbf{s}(\mathbf{x}) + \mathbf{r}(\mathbf{x})$.

The gradient of the differentiable function $\mathbf{s}$ at $\mathbf{x}$ is hence defined by $\nabla\mathbf{s}(\mathbf{x}) \triangleq \left[\nabla s_1^{\mathrm{T}}(x_{(1)}); \cdots; \nabla s_n^{\mathrm{T}}(x_{(n)})\right] \in \mathbb{R}^{n\times p}$ where $\nabla s_i(x_{(i)})$ denotes the gradient of s_i at x_i. For the nondifferentiable function $\mathbf{r}$, define $\tilde{\nabla}\mathbf{r}(\mathbf{x})$ as one of its subgradients at $\mathbf{x}$, $\tilde{\nabla}\mathbf{r}(\mathbf{x}) \triangleq [\tilde{\nabla} r_1^{\mathrm{T}}(x_{(1)}); \cdots \tilde{\nabla} r_n^{\mathrm{T}}(x_{(n)})] \in \mathbb{R}^{n\times p}$ where $\tilde{\nabla} r_i(x_{(i)})$ denotes one of the subgradients of r_i at x_i. Observe that the same row of $\mathbf{x}$, $\nabla\mathbf{s}(\mathbf{x})$, and $\tilde{\nabla}\mathbf{r}(\mathbf{x})$ corresponds to the same agent.

For a matrix A, we write its Frobenius norm as $\|A\|_{\mathrm{F}}$. For a matrix A and a positive semidefinite matrix G, define the G-matrix norm $\|A\|_G \triangleq \sqrt{\mathrm{trace}(A^{\mathrm{T}}GA)}$. For a matrix $A \in \mathbb{R}^{m\times n}$, let $\mathrm{null}\{A\} \triangleq \{x \in \mathbb{R}^n \big| Ax = 0\}$ be the null space of A and let $\mathrm{span}\{A\} \triangleq \{y \in \mathbb{R}^m \big| y = Ax, \forall x \in \mathbb{R}^n\}$ be the linear span of all the columns of A.

2. ALGORITHM DEVELOPMENT

This section first proposes PG-EXTRA to solve (1), then discusses two special cases of PG-EXTRA. When the local objective functions are differentiable (i.e., $r_i = 0$ for all i), PG-EXTRA reduces to EXTRA, the exact first-order algorithm proposed in [17]. When the smooth parts of the local cost functions vanish (i.e., $s_i = 0$ for all i), PG-EXTRA becomes a new decentralized proximal point algorithm, which we name as P-EXTRA.

2.1. PG-EXTRA

PG-EXTRA starts from an arbitrary initial solution $\mathbf{x}^0 \in \mathbb{R}^{n\times p}$. The next iterate $\mathbf{x}^1$ is updated through a gradient descent step (with respect to the differentiable function $\mathbf{s}$) followed by a proximal step (with respect to the nondifferentiable function $\mathbf{r}$)

$$\mathbf{x}^{\frac{1}{2}} = W\mathbf{x}^0 - \alpha\nabla\mathbf{s}(\mathbf{x}^0), \tag{4}$$

$$\mathbf{x}^1 = \arg\min_{\mathbf{x}} \ \mathbf{r}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^{\frac{1}{2}}\|_{\mathrm{F}}^2. \tag{5}$$

Here $\alpha \in \mathbb{R}$ is a positive step size, and $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ is a mixing matrix as we will discuss below. Then $\mathbf{x}^2, \mathbf{x}^3, \cdots$ are obtained through gradient steps on $\mathbf{s}$ followed by proximal steps on $\mathbf{r}$

$$\begin{aligned}\mathbf{x}^{k+1+\frac{1}{2}} = W\mathbf{x}^{k+1} + \mathbf{x}^{k+\frac{1}{2}} - \tilde{W}\mathbf{x}^k \\ - \alpha\left[\nabla\mathbf{s}(\mathbf{x}^{k+1}) - \nabla\mathbf{s}(\mathbf{x}^k)\right],\end{aligned} \tag{6}$$

$$\mathbf{x}^{k+2} = \arg\min_{\mathbf{x}} \ \mathbf{r}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^{k+1+\frac{1}{2}}\|_{\mathrm{F}}^2. \tag{7}$$

In (6) and (7), α and W are the same as those appearing in (4) and (5), while $\tilde{W} = [\tilde{w}_{ij}] \in \mathbb{R}^{n\times n}$ is a new mixing matrix. Observe that in (6), we first mix the previous iterates to obtain $W\mathbf{x}^{k+1} + \mathbf{x}^{k+\frac{1}{2}} - \tilde{W}\mathbf{x}^k$, then move along the direction $-\nabla\mathbf{s}(\mathbf{x}^{k+1})$, and finally move along the direction $\nabla\mathbf{s}(\mathbf{x}^k)$.

PG-EXTRA involves mixing neighboring iterates over the connected multi-agent network with two *mixing matrices* W and $\tilde{W}$. We impose the following assumptions on W and $\tilde{W}$.

Assumption 1 (Mixing matrices) *Consider a* connected network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ *consisting of a set of agents* $\mathcal{V} = \{1, 2, \cdots, n\}$ *and a set of undirected edges* $\mathcal{E}$. *An unordered pair* $(i, j) \in \mathcal{E}$ *if and only if agents* i *and* j *are directly connected. The mixing matrices* $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ *and* $\tilde{W} = [\tilde{w}_{ij}] \in \mathbb{R}^{n\times n}$ *satisfy*

1. *If* $i \neq j$ *and* $(i, j) \notin \mathcal{E}$, *then* $w_{ij} = \tilde{w}_{ij} = 0$.
2. $W = W^{\mathrm{T}}$, $\tilde{W} = \tilde{W}^{\mathrm{T}}$.
3. $\mathrm{null}\{W - \tilde{W}\} = \mathrm{span}\{\mathbf{1}\}$; $\mathrm{null}\{I - \tilde{W}\} \supseteq \mathrm{span}\{\mathbf{1}\}$.
4. $\tilde{W} \succ 0$ *and* $\frac{I+W}{2} \succcurlyeq \tilde{W} \succcurlyeq W$.

The first condition ensures the decentralized manner on the computing of PG-EXTRA, while last three conditions guarantee convergence of the algorithm. The mixing matrices W and $\tilde{W}$ diffuse information throughout the network and their roles are similar to those in EXTRA [17]. One can choose W as the mixing matrix in the decentralized (sub)gradient method [11], namely, W satisfies the first

two conditions, and has one eigenvalue being 1 while others being within $(-1, 1)$. After setting such a W, $(I + W)/2$ is a proper choice for $\tilde{W}$ that satisfies Assumption 1.

To see how the mixing matrices W and $\tilde{W}$ satisfying the first condition of Assumption 1 enable decentralized computation, we break down (4)–(7) to the individual agents and consider the operations of agent i. At the first iteration agent i runs

$$x_{(i)}^{\frac{1}{2}} = \sum_{j=1}^{n} w_{ij} x_{(j)}^0 - \alpha \nabla s_i(x_{(i)}^0), \tag{8}$$

$$x_{(i)}^1 = \arg\min_x \ r_i(x) + \frac{1}{2\alpha}\|x - x_{(i)}^{\frac{1}{2}}\|_2^2. \tag{9}$$

Since $w_{ij} = 0$ if $(i, j) \notin \mathcal{E}$ and $i \neq j$, to compute $\sum_{j=1}^n w_{ij} x_{(j)}^0$ agent i only needs to have iterates $x_{(j)}^0$ where agents j are its neighbors. The gradient descent term $-\alpha \nabla s_i(x_{(i)}^0)$ and the proximal step (9) are locally computable. At time $k + 2$, $k \geq 0$, agent i runs

$$x_{(i)}^{k+1+\frac{1}{2}} = \sum_{j=1}^{n} w_{ij} x_{(j)}^{k+1} + x_{(i)}^{k+\frac{1}{2}} - \sum_{j=1}^{n} \tilde{w}_{ij} x_{(j)}^k - \alpha \left[\nabla s_i(x_{(i)}^{k+1}) - \nabla s_i(x_{(i)}^k)\right], \tag{10}$$

$$x_{(i)}^{k+2} = \arg\min_x \ r_i(x) + \frac{1}{2\alpha}\|x - x_{(i)}^{x+1+\frac{1}{2}}\|_2^2. \tag{11}$$

Because $w_{ij} = 0$ and $\tilde{w}_{ij} = 0$ if $(i, j) \notin \mathcal{E}$ and $i \neq j$, the mixing term $\sum_{j=1}^n w_{ij} x_{(j)}^{k+1} + x_{(i)}^{k+\frac{1}{2}} - \sum_{j=1}^n \tilde{w}_{ij} x_{(j)}^k$ can be computed using local and neighboring iterates. For a neighboring agent j, agent i requires its latest iterate $x_{(j)}^{k+1}$, while the previous iterate $x_{(j)}^k$ has already been collected at the preceding iteration. Similar to (8) and (9), the gradient descent-ascent term $-\alpha[\nabla s_i(x_{(i)}^{k+1}) - \nabla s_i(x_{(i)}^k)]$ in (10) and the proximal step (11) are also locally computable.

PG-EXTRA is outlined in **Algorithm 1**.

Algorithm 1: PG-EXTRA

Set mixing matrices $W \in \mathbb{R}^{n\times n}$ and $\tilde{W} \in \mathbb{R}^{n\times n}$;
Choose step size $\alpha > 0$;

1. For each agent i, pick any initial iterate $x_{(i)}^0 \in \mathbb{R}^n$ and compute

$$x_{(i)}^{\frac{1}{2}} = \sum_{j=1}^{n} w_{ij} x_{(j)}^0 - \alpha \nabla s_i(x_{(i)}^0),$$

$$x_{(i)}^1 = \arg\min_x \ r_i(x) + \tfrac{1}{2\alpha}\|x - x_{(i)}^{\frac{1}{2}}\|_2^2.$$

2. **for** $k = 0, 1, \cdots$, for each agent i, **do**

$$x_{(i)}^{k+1+\frac{1}{2}} = \sum_{j=1}^{n} w_{ij} x_{(j)}^{k+1} + x_{(i)}^{k+\frac{1}{2}} - \sum_{j=1}^{n} \tilde{w}_{ij} x_{(j)}^k - \alpha \left[\nabla s_i(x_{(i)}^{k+1}) - \nabla s_i(x_{(i)}^k)\right],$$

$$x_{(i)}^{k+2} = \arg\min_x \ r_i(x) + \tfrac{1}{2\alpha}\|x - x_{(i)}^{x+1+\frac{1}{2}}\|_2^2.$$

end for

2.2. Special Cases: EXTRA and P-EXTRA

When the aggregate objective function $\mathbf{f}$ has simpler forms, PG-EXTRA also reduces to simpler ones.

If $\mathbf{r} = 0$, observe that in (4) and (5) we have $\mathbf{x}^1 = \mathbf{x}^{\frac{1}{2}}$

$$\mathbf{x}^1 = W\mathbf{x}^0 - \alpha \nabla \mathbf{s}(\mathbf{x}^0). \tag{12}$$

Similarly, in (6) and (7) we have $\mathbf{x}^{k+2} = \mathbf{x}^{k+1+\frac{1}{2}}$ and

$$\mathbf{x}^{k+2} = W\mathbf{x}^{k+1} + \mathbf{x}^{k+1} - \tilde{W}\mathbf{x}^k - \alpha \left[\nabla \mathbf{s}(\mathbf{x}^{k+1}) - \nabla \mathbf{s}(\mathbf{x}^k)\right]. \tag{13}$$

Indeed, the updates (12) and (13) are identical to those in EXTRA, the exact first-order algorithm that solves (1), with the local objective functions being differentiable [17]. In this sense, PG-EXTRA is an extension of EXTRA from decentralized differentiable consensus optimization to the nondifferentiable regime.

If $\mathbf{s} = 0$, we have a new algorithm named as P-EXTRA which stands for proximal-EXTRA. To see how the name comes, observe that to compute $\mathbf{x}^1$ the updates are

$$\mathbf{x}^{\frac{1}{2}} = W\mathbf{x}^0, \tag{14}$$

$$\mathbf{x}^1 = \arg\min_{\mathbf{x}} \ \mathbf{r}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^{\frac{1}{2}}\|_{\mathrm{F}}^2. \tag{15}$$

Here (14) simply mixes $\mathbf{x}^0$ with W to obtain $\mathbf{x}^{\frac{1}{2}}$ without the gradient descent operation, whereas (15) is a proximal step. And to compute $\mathbf{x}^{k+2}$ the updates are

$$\mathbf{x}^{k+1+\frac{1}{2}} = W\mathbf{x}^{k+1} + \mathbf{x}^{k+\frac{1}{2}} - \tilde{W}\mathbf{x}^k, \tag{16}$$

$$\mathbf{x}^{k+2} = \arg\min_{\mathbf{x}} \ \mathbf{r}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^{k+1+\frac{1}{2}}\|_{\mathrm{F}}^2. \tag{17}$$

Similar to (14) and (15), (16) and (17) are pure mixing and proximal steps, respectively.

P-EXTRA is outlined in **Algorithm 2**.

Algorithm 2: P-EXTRA

Set mixing matrices $W \in \mathbb{R}^{n\times n}$ and $\tilde{W} \in \mathbb{R}^{n\times n}$;
Choose step size $\alpha > 0$;

1. For each agent i, pick any initial iterate $x_{(i)}^0 \in \mathbb{R}^n$ and compute

$$x_{(i)}^{\frac{1}{2}} = \sum_{j=1}^{n} w_{ij} x_{(j)}^0,$$

$$x_{(i)}^1 = \arg\min_x \ r_i(x) + \tfrac{1}{2\alpha}\|x - x_{(i)}^{\frac{1}{2}}\|_2^2.$$

2. **for** $k = 0, 1, \cdots$, for each agent i, **do**

$$x_{(i)}^{k+1+\frac{1}{2}} = \sum_{j=1}^{n} w_{ij} x_{(j)}^{k+1} + x_{(i)}^{k+\frac{1}{2}} - \sum_{j=1}^{n} \tilde{w}_{ij} x_{(j)}^k,$$

$$x_{(i)}^{k+2} = \arg\min_x \ r_i(x) + \tfrac{1}{2\alpha}\|x - x_{(i)}^{x+1+\frac{1}{2}}\|_2^2.$$

end for

3. CONVERGENCE ANALYSIS

To establish convergence and rate of convergence for PG-EXTRA and P-EXTRA, we make two additional assumptions.

Assumption 2 (Convexity and Lipschitz differentiability) *For any i, functions s_i and r_i are proper closed convex and the differentiable function s_i satisfies*

$$\|\nabla s_i(x_a) - \nabla s_i(x_b)\| \leq L_{s_i}\|x_a - x_b\|_2, \quad \forall x_a, x_b \in \mathbb{R}^p,$$

where $L_{s_i} > 0$ is constant.

Following Assumption 2, function $\mathbf{f}(\mathbf{x}) = \mathbf{s}(\mathbf{x}) + \mathbf{r}(\mathbf{x})$ is proper closed convex, and $\nabla \mathbf{s}$ is Lipschitz continuous

$$\|\nabla \mathbf{s}(\mathbf{x}_a) - \nabla \mathbf{s}(\mathbf{x}_b)\|_{\mathrm{F}} \leq L_{\mathbf{s}}\|\mathbf{x}_a - \mathbf{x}_b\|_{\mathrm{F}}, \quad \forall \mathbf{x}_a, \mathbf{x}_b \in \mathbb{R}^{n\times p},$$

with constant $L_{\mathbf{s}} = \max_i\{L_{s_i}\}$.

Assumption 3 (Solution existence) *Problem* (1) *has a set of optimal solutions $\mathcal{X}^* \neq \emptyset$ which is a convex set.*

We first give a lemma that gives the first-order optimality conditions of (1). This and other proofs can be found in [24].

Lemma 1 ***(First-order optimality conditions)*** *Given mixing matrices W and $\tilde{W}$, define $U = (\tilde{W} - W)^{1/2}$ by letting $U \triangleq VS^{1/2}V^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ where $VSV^{\mathrm{T}} = \tilde{W} - W$ is the economical-form singular value decomposition. Then, under Assumptions 1–3, $\mathbf{x}^*$ is consensual and $x^*_{(1)} \equiv x^*_{(2)} \equiv \cdots \equiv x^*_{(n)}$ is optimal to problem* (1) *if and only if there exists $\mathbf{q}^* = U\mathbf{p}$ for some $\mathbf{p} \in \mathbb{R}^{n\times p}$ such that*

$$U\mathbf{q}^* + \alpha\left(\nabla \mathbf{s}(\mathbf{x}^*) + \tilde{\nabla}\mathbf{r}(\mathbf{x}^*)\right) = \mathbf{0}, \tag{18}$$

$$U\mathbf{x}^* = \mathbf{0}. \tag{19}$$

Suppose that $\mathbf{x}^*$ and $\mathbf{q}^*$ satisfy the optimality conditions (18) and (19). Introduce an auxiliary sequence $\mathbf{q}^k \triangleq \sum_{t=0}^{k} U\mathbf{x}^t$. In light of (18) and (19), define *optimality residuals* as $\|U\mathbf{q}^k + \alpha(\nabla \mathbf{s}(\mathbf{x}^k) + \tilde{\nabla}\mathbf{r}(\mathbf{x}^{k+1}))\|_{\mathrm{F}}^2$ and $\|U\mathbf{x}^k\|_{\mathrm{F}}^2$; the former is the violation to the first-order optimality of (1) while the latter is the violation of consensus. For PG-EXTRA, the following theorem shows convergence of $\mathbf{x}^k$ to $\mathbf{x}^*$, as well as the $o(1/k)$ rates of the optimality residuals.

Theorem 1 *Under Assumptions 1–3, if the step size satisfies $0 < \alpha < 2\lambda_{\min}(\tilde{W})/L_{\mathbf{s}}$), then the iterate $\mathbf{x}^k$ generated by PG-EXTRA converges to an optimal $\mathbf{x}^*$ and the running-best optimality residuals have rates*

$$\min_{t\le k}\left\{\|U\mathbf{q}^t + \alpha(\nabla \mathbf{s}(\mathbf{x}^t) + \tilde{\nabla}\mathbf{r}(\mathbf{x}^{t+1}))\|_{\mathrm{F}}^2\right\} = o\left(\frac{1}{k}\right), \tag{20}$$

$$\min_{t\le k}\left\{\|U\mathbf{x}^t\|_{\mathrm{F}}^2\right\} = o\left(\frac{1}{k}\right). \tag{21}$$

The convergence of P-EXTRA follows from that of PG-EXTRA directly. Since P-EXTRA considers a simpler case in which the differentiable part $\mathbf{s}(\mathbf{x}) = \mathbf{0}$, it allows arbitrary positive step size and has $o(1/k)$ rates in terms of the optimality conditions instead of their running-bests. See the theorem below.

Theorem 2 *Under Assumptions 1–3 and $\mathbf{s}(\mathbf{x}) = 0$, for any step size $\alpha > 0$, the iterate $\mathbf{x}^k$ generated by P-EXTRA converges to an optimal $\mathbf{x}^*$ and the optimality residuals have rates*

$$\|U\mathbf{q}^k + \alpha\tilde{\nabla}\mathbf{r}(\mathbf{x}^k)\|_{\mathrm{F}}^2 = o\left(\frac{1}{k}\right), \tag{22}$$

$$\|U\mathbf{x}^k\|_{\mathrm{F}}^2 = o\left(\frac{1}{k}\right). \tag{23}$$

4. NUMERICAL EXPERIMENTS

Numerical experiments are conducted over a connected network consisting of $n = 10$ agents and 18 bidirectional edges, as shown in Fig. 1. We simulate a decentralized compressive sensing problem [4, 20]. Each agent i holds its own measurement equation $y_{(i)} = \mathbf{M}_{(i)}x + e_{(i)}$, where $y_{(i)} \in \mathbb{R}^{m_i}$ is measured data, $\mathbf{M}_{(i)} \in \mathbb{R}^{m_i\times p}$ is measurement matrix, $x \in \mathbb{R}^p$ is unknown sparse signal, and $e_{(i)} \in \mathbb{R}^{m_i}$ is unknown noise. The goal of the agents is to collaboratively estimate the sparse signal x. To find x, the compressive sensing theory suggests to solve an ℓ_1 regularized least squares problem in the form

$$\underset{x}{\text{minimize}} \sum_{i=1}^{n} s_i(x) + \sum_{i=1}^{n} r_i(x),$$

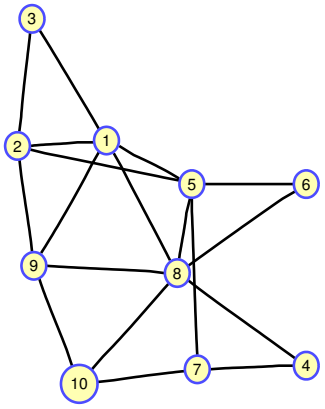

Fig. 1. The underlying graph for numerical experiments is a connected network with $n = 10$ agents and 18 bidirectional edges.

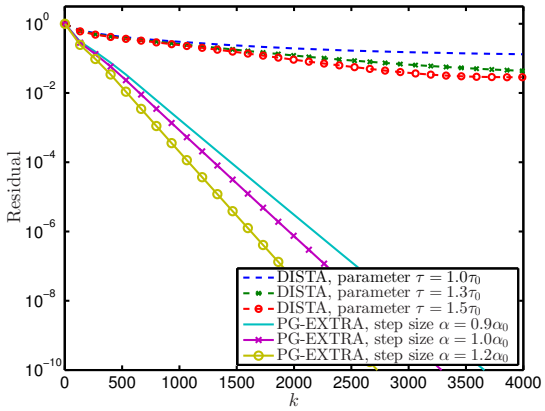

Fig. 2. The normalized optimal residual $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathrm{F}}/\|\mathbf{x}^0 - \mathbf{x}^*\|_{\mathrm{F}}$. For PG-EXTRA, $\alpha_0 = 0.82193$ is the critical step size given in Theorem 1; $\tau_0 = 1$ is the parameter of DISTA.

where $s_i(x) = \frac{1}{2}\|\mathbf{M}_{(i)}x - y_{(i)}\|_2^2$, $r_i(x) = \lambda_{(i)}\|x\|_1$, and $\lambda_{(i)}$ is the regularization parameter on agent i. In the experiments, each agent i holds $m_i = 3$ measurements, its regularization parameter $\lambda_i = \frac{1}{n}$. The sparse signal x has dimension $p = 50$ and its 80% of elements are zero. The entries of the measurement matrices $\mathbf{M}_{(i)}$ and the nonzero elements of the signal x are generated following i.i.d. Gaussian distribution with mean 0 and standard deviation 1, and i.i.d. Laplace distribution with mean 0 and diversity 1, respectively. The elements of the noise vectors $e_{(i)}$ are generated following i.i.d. Gaussian distribution with mean 0 and standard deviation 0.1.

The numerical results are illustrated in Fig. 2. We compare PG-EXTRA with DISTA [25], which is a decentralized version of the iterative soft thresholding algorithm (ISTA) [26]. We use the normalized optimal residual $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathrm{F}}/\|\mathbf{x}^0 - \mathbf{x}^*\|_{\mathrm{F}}$ as performance metric. PG-EXTRA demonstrates fast convergence to the optimal solution given proper step size. DISTA converges much slower since it is essentially a proximal version of decentralized gradient descent [11], which is disadvantageous in convergence speed.

5. REFERENCES

[1] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*, Princeton University Press, 2009

[2] K. Zhou and S. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," IEEE Transactions on Robotics, vol. 27, pp. 678–695, 2010

[3] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc WSNs with noisy links - Part I: Distributed estimation of deterministic signals," IEEE Transactions on Signal Processing, vol. 56, pp. 350–364, 2008

[4] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," IEEE Transactions on Signal Processing, vol. 58, pp. 3816–3827, 2010

[5] V. Kekatos and G. Giannakis, "Distributed robust power system state estimation," IEEE Transactions on Power Systems, vol. 28, pp. 1617–1626, 2013

[6] G. Giannakis, V. Kekatos, N. Gatsis, S. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," IEEE Signal Processing Magazine, vol. 30, pp. 107–128, 2013

[7] P. Forero, A. Cano, and G. Giannakis, "Consensus-based distributed support vector machines," Journal of Machine Learning Research, vol. 11, pp. 1663–1707, 2010

[8] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," IEEE Transactions on Knowledge and Data Engineering, vol. 25, pp. 2483–2493, 2013

[9] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," In: Proceedings of IEEE International Workshop on Signal Processing Advances for Wireless Communications, 2005

[10] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," IEEE Transactions on Signal Processing, vol. 62, pp. 1750–1761, 2014

[11] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," IEEE Transactions on Automatic Control, vol. 54, pp. 48–61, 2009

[12] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," IEEE Transactions on Automatic Control, vol. 57, pp. 592–606, 2012

[13] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2014

[14] T. Chang, M. Hong, and X. Wang, "Multiagent distributed large-scale optimization by inexact consensus alternating direction method of multipliers," In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2014

[15] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," IEEE Transactions on Automatic Control, To Appear

[16] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," Manuscript

[17] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," Manuscript

[18] H. Eiselt and V. Marianov, *Foundations of Location Analysis*, Springer, 2011

[19] H. Lopuhaa and P. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," The Annals of Statistics, vol. 19, pp. 229–248, 1991

[20] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," IEEE Transactions on Signal Processing, vol. 58, pp. 5262–5276, 2010

[21] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," IEEE Journal of Selected Topics in Signal Processing, vol. 7, pp. 221–229, 2013

[22] T. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," IEEE Transactions on Automatic Control, vol. 59, pp. 1524–1538, 2014

[23] C. Pang, "Set intersection problems: Supporting hyperplanes and quadratic programming," Mathematical Programming, To Appear

[24] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized nondifferentiable optimization," Manuscript available at `http://home.ustc.edu.cn/~qingling/papers/PG-EXTRA.pdf`

[25] C. Ravazzi, S. M. Fosson, and E. Magli, "Distributed soft thresholding for sparse signal recovery," In: Proceedings of IEEE Global Communications Conference, 2013

[26] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," Communications on pure and applied mathematics, vol. 57, pp. 1413–1457, 2004