NPGA: A Unified Algorithmic Framework for Decentralized Constraint-Coupled Optimization

Jingwang Li and Housheng Su

Abstract—This work focuses on a class of general decentralized constraint-coupled optimization problems. We propose a novel nested primal–dual gradient algorithm (NPGA), which can achieve linear convergence under the weakest known condition, and its theoretical convergence rate surpasses all known results. More important, NPGA serves not only as an algorithm but also as a unified algorithmic framework, encompassing various existing algorithms as special cases. By designing different network matrices, we can derive numerous versions of NPGA and analyze their convergence by leveraging the convergence results of NPGA conveniently, thereby enabling the design of more efficient algorithms. Finally, we conduct numerical experiments to compare the convergence rates of NPGA and existing algorithms, providing empirical evidence for the superior performance of NPGA.

Index Terms—Constraint-coupled optimization, linear convergence, nested primal–dual gradient algorithm (NPGA), unified algorithmic framework.

I. INTRODUCTION

Consider a networked system composed of \( n \) agents, where the network topology is represented by an undirected graph \( G = (\mathcal{N}, \mathcal{E}) \), where \( \mathcal{N} = \{1, \ldots, n\} \) and \( \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \) denote the sets of nodes and edges, respectively. Specifically, \((i, j) \in \mathcal{E}\) if agents \( i \) and \( j \) can communicate with each other. In this work, we consider the following decentralized constraint-coupled optimization problem:

\[
\min_{x_i \in \mathbb{R}^{d_i}} \sum_{i=1}^{n} f_i(x_i) + g_i(x_i) + h \left( \sum_{i=1}^{n} A_i x_i \right) \tag{P1}
\]

where \( f_i : \mathbb{R}^{d_i} \to \mathbb{R}, g_i : \mathbb{R}^{d_i} \to \mathbb{R} \cup \{+\infty\}, \) and \( A_i \in \mathbb{R}^{m \times d_i} \) are completely private to agent \( i \). This means that agent \( i \) cannot share this information with other agents in the system. On the other hand, \( h : \mathbb{R}^{P} \to \mathbb{R} \cup \{+\infty\} \) is public and known to all agents. Without loss of generality, we assume that the solution set of (P1) is nonempty.

Different from the classical decentralized unconstrained optimization problem

\[
\min_{x \in \mathbb{R}^{m}} \sum_{i=1}^{n} \phi_i(x). \tag{P2}
\]

(P1) introduces an additional cost function \( h \) that couples the decision variables of all agents. Notice that we can reformulate (P1) as a new problem with a globally coupled equality constraint \( \sum_{i=1}^{n} A_i x_i = y \) by introducing a relaxed variable \( y \), which is the reason we refer to it as a decentralized constraint-coupled optimization problem [1]. Many real-world optimization problems can be modeled or reformulated as (P1), including vertically distributed regression and vertical federated learning [2, 3]; distributed basis pursuit [4]; and distributed resource allocation [5, 6, 7].

In this work, we aim to answer the following two questions.

1) Can we develop linearly convergent distributed algorithms for (P1) with weaker convergence conditions and faster convergence rates?

2) Can we design a unified algorithmic framework for (P1)?

Regarding the first question, while there have been some linearly convergent distributed algorithms proposed for problems similar to (P1), most of them can only handle specific cases of (P1). This means that stronger conditions need to be imposed on (P1) to make it solvable by these algorithms. These conditions include having \( g_i = 0 \) and \( h \) being an indicator function corresponding to the equality constraint [2, 5, 6, 7, 8, 9, 10, 11]; \( A_i = 1 \) [8]; \( A_i = I \) [5, 6, 7, 9]; or \( A_i \) has full row rank [2, 10]. However, these conditions are often too restrictive and cannot be satisfied by most real-world problems. For example, in the context of vertical federated learning, the dimensions of \( A_i \) correspond to the numbers of samples and features that agent \( i \) owns, respectively. However, the number of samples is usually much larger than that of features, which means that the conditions about \( A_i \) mentioned earlier cannot be met. Therefore, there is a need for linearly convergent distributed algorithms with weaker conditions. Some recent progress has been made in this direction [11, 12]. In [11], an implicit tracking-based distributed augmented primal–dual gradient dynamics (IDEA) algorithm is proposed, which can converge without the need for the strong convexity of \( f_i \) and achieve linear convergence under the condition that \( g_i = 0 \).
an indicator function corresponding to the equality constraint, and \(A = [A_1, \ldots, A_n]\) has full row rank, which is much weaker than the aforementioned conditions about \(A_i\). On the other hand, Alghunaim et al. [12] focus solely on linear convergence, and the dual consensus proximal algorithm (DCPA) is proposed, which achieves linear convergence under the condition that \(g_i = 0\) and \(A\) has full row rank or \(h\) is smooth. However, there are some drawbacks to both IDEA and DCPA, which are as follows:

1) Although the linear convergence rate of IDEA has been established, as a continuous-time algorithm, it needs to be discretized for practical implementation. However, the upper bounds of the discretized step-sizes and the corresponding convergence rate of its discretization are not clear.

2) For DCPA, the linear convergence rate is obtained by inevitably imposing a smaller bound on one of its step sizes, which leads to a looser convergence rate.

3) Many adapt-then-combine (ATC) algorithms have been proposed for (P2) [13], [14], [15], and it has been shown that ATC algorithms have additional favorable properties compared to combine-then-apply (CTA) ones,\(^1\) such as the ability to use uncoordinated step-sizes or step-sizes as large as their centralized counterparts [13], [15]. However, both IDEA and DCPA are CTA algorithms; thus, it remains an open problem whether ATC algorithms can linearly solve (P1) under the same or weaker conditions as CTA ones and if they are superior in other aspects.

In terms of the second question, several unified algorithmic frameworks have been proposed for the decentralized unconstrained optimization problem (P2) [17], [18], [19], [20]. However, to the best of our knowledge, there is no such framework for (P1). Given that there are only two algorithms capable of linearly solving (P1) under weaker conditions, it becomes crucial to develop a unified algorithmic framework that enables the convenient design of new algorithms and facilitates the analysis of their convergences.

The major contributions of this article are summarized as follows.

1) Regarding the second question, we propose a novel nested primal–dual gradient algorithm (NPCA) for (P1), which is a unified algorithmic framework. NPCA involves the configuration of three network matrices, allowing it to encompass existing algorithms, such as DCPA and the dual coupled diffusion algorithm (DCDA) [10], as special cases. To the best of our knowledge, this is the first unified algorithmic framework for solving (P1) and its variants. By designing new network matrices within NPCA, we can generate novel distributed algorithms, including both ATC and CTA algorithms, for (P1). The convergence analysis of these new algorithms can be conveniently conducted by leveraging the convergence results of NPCA.

2) For the first question, we prove that NPCA achieves linear convergence under the weakest known condition: \(g_i = 0\) and \(A = [A_1, \ldots, A_n]\) has full row rank, or \(h\) is smooth. As mentioned before, DCPA can achieve linear convergence under the same condition. However, DCPA is merely a special case of NPCA, and its convergence results can be fully encompassed by those of NPCA. Furthermore, we prove that certain ATC variants of NPCA exhibit larger upper bounds of step sizes and tighter convergence rates compared to DCPA. Numerical experiments also demonstrate that certain versions of NPCA have much faster convergence rates than DCPA in terms of both the number of iterations and communication rounds, which correspond to computational and communication costs, respectively.

3) For the special case of (P1) with \(g_i = 0\) and \(h\) representing an indicator function corresponding to the equality constraint, we derive a weaker condition concerning the network matrices of NPCA to ensure linear convergence and obtain a tighter convergence rate. This extension significantly expands the range of network matrix choices within NPCA. As a special case of NPCA, the convergence rate of DCPA can also be improved. Additionally, we show that certain ATC versions of NPCA can employ step sizes as large as the centralized version of NPCA, resulting in a larger stability region compared to other versions of NPCA, including DCPA.

II. PRELIMINARIES

In this section, we introduce the notations and lemmas utilized throughout this article.

**Notations:** \(1_n\) and \(0_n\) represent the vectors of \(n\) ones and zeros, respectively. \(I_n\) denotes the \(n \times n\) identity matrix, and \(0_{m \times n}\) denotes the \(m \times n\) zero matrix. Notice that if the dimensions can be inferred from the context, we would not explicitly indicate them. For a positive semidefinite matrix \(B \in \mathbb{R}^{n \times n}\), we define \(\|x\|^2_B = x^\top B x\). Given two symmetric matrices \(A \in \mathbb{R}^{n \times n}\) and \(B \in \mathbb{R}^{n \times n}\), \(A \succ B\) (or \(A \geq B\)) indicates that \(A - B\) is positive definite (positive semidefinite). For \(A \in \mathbb{R}^{n \times n}\), \(\eta(A)\), \(\eta^2(A)\), and \(\eta(A)\) denote the smallest, smallest nonzero, and largest eigenvalues of \(A\), respectively. For \(B \in \mathbb{R}^{m \times n}\), \(\sigma(B)\), \(\sigma_i(B)\), and \(\mathcal{S}(B)\) represent the smallest nonzero, \(i\)th smallest, and largest singular values of \(B\), respectively. \(\text{Null}(B)\) and \(\text{Col}(B)\) denote the null space and the column space of \(B\), respectively. For a vector \(v \in \mathbb{R}^n\), \(\text{Span}(v)\) denotes its span. For a function \(f : \mathbb{R}^m \to \mathbb{R}\), \(\partial f(x)\) denotes any subgradient of \(f\) at \(x\), \(\hat{f}(x)\) denotes the subdifferential of \(f\) at \(x\), which is the set of all subgradients of \(f\) at \(x\), and \(\text{prox}_{\alpha f}(x) = \arg \min_{y} f(y) + \frac{1}{2\alpha} \|y - x\|^2\) represents the proximal operator of \(f\) with step-size \(\alpha\). Finally, \(\text{diag}(\cdot)\) denotes the (block) diagonal matrix.

**Lemma 1 (See [21]):** Let \(f : \mathbb{R}^m \to \mathbb{R}\) be a \(\mu\)-strongly convex and \(l\)-smooth function, it holds that

\[
\begin{align*}
\langle \nabla f(x), y - x \rangle &\geq f(y) - f(x) + \frac{\mu}{2} \|x - y\|^2 \\
\langle \nabla f(x) - \nabla f(y), y - x \rangle &\geq \frac{1}{2} \|x - y\|^2 \\
\|\nabla f(x) - \nabla f(y)\|^2 &\leq \|\nabla f(x) - \nabla f(y)\|^2 (w - y) \\
\|\nabla f(x) - \nabla f(y)\|^2 &\leq l(\nabla f(x) - \nabla f(y)) (w - y)
\end{align*}
\]

\(^1\)There are two types of information diffusion strategies over networks: 1) CTA and 2) ATC [16]. Accordingly, existing distributed optimization algorithms can be categorized as CTA or ATC algorithms based on their information diffusion strategies.
for any \( x, y \in \mathbb{R}^m \).

**Lemma 2:** Given two matrices \( M \in \mathbb{R}^{m \times n} \) and \( H \in \mathbb{R}^{n \times n} \), where \( H \) is symmetric, it holds that \( \sigma_i(MH) \leq \sigma_i(M)\sigma_i(H) \), \( i = 1, \ldots, n \).

**Proof:** See the full version of this work [22]. ■

### III. Algorithm Design

Let \( d = \sum_{i=1}^{n} d_i \) and \( x = [x_1^T, \ldots, x_n^T]^T \in \mathbb{R}^d \), (P1) can be reformulated as

\[
\min_{x \in \mathbb{R}^d} f(x) + g(x) + h(Ax)
\]

where \( f(x) = \sum_{i=1}^{n} f_i(x_i) \), \( g(x) = \sum_{i=1}^{n} g_i(x_i) \), and \( A = [A_1, \ldots, A_n] \in \mathbb{R}^{p \times d} \). Let \( f(x) = f(x) + g(x) \), and introduce a slack variable \( y \in \mathbb{R}^p \); then, we can rewrite (P3) as

\[
\min_{x \in \mathbb{R}^d, y \in \mathbb{R}^p} f(x) + h(y)
\]

s.t. \( Ax - y = 0 \). (1)

The corresponding saddle-point problem is given by

\[
\max_{\lambda \in \mathbb{R}^p} \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^p} f(x) + h(y) + \lambda^\top (Ax - y)
\]

\[
= \max_{\lambda \in \mathbb{R}^p} \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^p} f(x) + \lambda^\top Ax + \min_{y \in \mathbb{R}^p} h(y) - \lambda^\top y
\]

\[
= \max_{\lambda \in \mathbb{R}^p} \min_{x \in \mathbb{R}^d} f(x) + \lambda^\top Ax - h^*(\lambda)
\]

(2)

where \( \lambda \) is the Lagrange multiplier and \( L \) is the Lagrangian function of (P3). Therefore, (P3) is equivalent to the saddle-point problem

\[
\min_{x \in \mathbb{R}^d} \max_{\lambda \in \mathbb{R}^p} f(x) + \lambda^\top Ax - h^*(\lambda)
\]

which implies that if \( (x^*, \lambda^*) \) solves (3), then \( x^* \) solves (P3). Throughout the article, we assume the following assumptions hold.

**Assumption 1:** There exists at least a saddle point to \( L \).

**Remark 1:** Recall that the solution set of (P3) is nonempty, thus Assumption 1 holds if the strong duality holds for (P3), which can be guaranteed by certain constraint qualifications. A popular one is Slater’s condition [31], which, when applied to (1), states that the strong duality holds for (P3) if (P3) convex and there exists a point \( x \in \text{relint}(D(g)) \) such that \( Ax \in \text{relint}(D(h)) \), where \( D(\cdot) \) denotes the domain of a function and \( \text{relint}(\cdot) \) represents the relative interior [31] of a set. Notice that \( \text{relint}(\mathbb{R}^p) = \mathbb{R}^p \); hence, Assumption 1 naturally holds if \( D(g) = \mathbb{R}^d \) and \( D(h) = \mathbb{R}^p \), which is typically satisfied by most practical optimization problems (see examples in Section V).

**Assumption 2:** The network topology among agents is an undirected and connected graph.

**Assumption 3:** The cost functions satisfy the following.

1) \( f_i \) is \( \mu_i \)-strongly convex and \( l_i \)-smooth, where \( \mu_i \) and \( l_i \) are positive constants \( \forall i \in N \).
2) \( g_i \) and \( h \) are convex but not necessarily smooth \( \forall i \in N \).

According to Assumption 3, we can conclude that \( f \) is \( \mu \)-strongly convex and \( l \)-smooth, where \( \mu = \min_{i \in N} \mu_i \) and \( l = \max_{i \in N} l_i \).

As an efficient algorithm for solving saddle-point problems, the primal–dual (proximal) gradient algorithm (PGA) has various versions [32], [33], [34]. In this article, we consider the one proposed in [34], which is given as

\[
x^{k+1} = \text{prox}_{\alpha g} (x^k - \alpha (\nabla f(x^k) + A^\top \lambda^k))
\]

\[
\lambda^{k+1} = \text{prox}_{\beta h} (\lambda^k + \beta A(x^{k+1} + \theta(x^{k+1} - x^k)))
\]

(4)

where \( \alpha, \beta > 0 \) and \( \theta \geq 0 \). However, we cannot directly employ PGA to solve (P3) in a decentralized manner since the update of \( \lambda \) requires the globally coupled term \( Ax^{k+1} = \sum_{i=1}^{n} A_i x_i^{k+1} \). To overcome this challenge, we propose an efficient decentralized variant of PGA, called NPGA, based on the dual perspective.

We begin by observing that the dual function of (P3) can be decomposed as

\[
\phi(\lambda) = \inf_{x \in \mathbb{R}^d} f(x) + g(x) + \lambda^\top Ax - h^*(\lambda)
\]

\[
= \inf_{i=1}^{n} \left( \inf_{x_i \in \mathbb{R}^{d_i}} f_i(x_i) + g_i(x_i) + \lambda^\top A_i x_i \right) - h^*(\lambda)
\]

\[
= \sum_{i=1}^{n} \phi_i(\lambda) - h^*(\lambda)
\]

where \( \phi_i(\lambda) = \inf_{x_i \in \mathbb{R}^{d_i}} f_i(x_i) + g_i(x_i) + \lambda^\top A_i x_i \). Consequently, we can reformulate the dual problem of (P3) as

\[
\max_{\lambda \in \mathbb{R}^p} \sum_{i=1}^{n} \phi_i(\lambda) - h^*(\lambda)
\]

(5)

which corresponds to the classical decentralized unconstrained optimization problem (P2) (with an extra term \( h^*(\lambda) \)). It is well known that (P2) has been widely studied in recent years and numerous effective algorithms has been proposed to solve it, including DIGing [23], [24]; exact first-order algorithm (EXTRA) [25]; decentralized linearized alternating direction method of multipliers (DLM) [26]; proximal primal–dual diffusion (P2D2) [27]; augmented distributed gradient methods (Aug-DGM) [13], [14]; ATC tracking [28], [29]; exact diffusion [30]; and network independent step-size (NIDS) [15].

Note that Assumption 1 implies that the strong duality holds for (P3); thus, we can employ numerous distributed unconstrained optimization algorithms to solve (5). Once (5) is solved, solving (P3) becomes straightforward. The question lies in selecting the most appropriate algorithm to solve (5) among the existing distributed unconstrained optimization algorithms. Although any algorithm is theoretically feasible, the performance of different algorithms may vary significantly depending on the specific scenario. Therefore, we are interested in a general primal–dual algorithm framework called the proximal unified decentralized algorithm (PUDA) [17]. PUDA offers flexibility by allowing the selection of different network matrices \( B \in \mathbb{R}^{n \times n} \), \( C \in \mathbb{R}^{n \times n} \), and \( D \in \mathbb{R}^{n \times n} \), which are associated with the network topology among the agents. By appropriately choosing these matrices, PUDA can recover many existing distributed unconstrained optimization algorithms, as shown in Table 1. The detailed derivation of the equivalence between PUDA and existing algorithms can be found in [17].
TABLE I
EXISTING DISTRIBUTED UNCONSTRAINED OPTIMIZATION ALGORITHMS UNIFIED BY PUDA [17]

| Algorithm       | $B^2$ | $C$          | $D$          | Communication rounds |
|-----------------|-------|--------------|--------------|----------------------|
| CTA algorithms  |       | $\frac{1}{2}(I-W)^2$ | $I-W^2$      | 2                    |
| EXTRA [25]      | $\frac{1}{2}(I-W)$ | $I$          | 1            | 1                    |
| DLM [26]        | $\frac{1}{2}(I-W)$ | $I$          | 1            | 1                    |
| P2D2 [27]       | $\frac{1}{2}(I-W)$ | $I$          | 1            | 1                    |
| ATC algorithms  |       | $\frac{1}{2}(I-W)^2$ | $I-W^2$      | 2                    |
| Avg.DGM [13, 14]|       | $I-W^2$      | $\frac{1}{2}(I-W)$ | 2                    |
| ATC tracking [28, 29]|       | $I-W^2$      | $\frac{1}{2}(I-W)$ | 2                    |
| Exact diffusion [30]|       | $\frac{1}{2}(I-W)$ | $I-W$        | 1                    |
| NDS [15]        | $\frac{1}{2}(I-W)$ | $I-W$        | 1            | 1                    |

$L \in \mathbb{R}^{n \times n}$ and $W \in \mathbb{R}^{n \times n}$ are the Laplacian and mixing matrices associated with $\gamma$, respectively, and $c > 0$ is a tunable constant. The last column denotes the number of communication rounds of the corresponding algorithm at each iteration.

**Remark 2:** For an undirected and connected graph, its corresponding mixing matrix $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ satisfies the following [25].

1. $w_{ij} > 0$ if $i = j$ or $(i, j) \in \mathcal{E}$, otherwise $w_{ij} = 0$.
2. $W = W^T$.
3. $\text{Null}(I-W) = \text{Span}(1_n)$.

Given the aforementioned three properties, we can easily verify that $-I < W < I$ [25]. There are various methods to construct a mixing matrix $W$ satisfying these properties, and one commonly used approach is the Laplacian method [35]

$$W = I - \frac{L}{\tau}$$

where $\tau = \max_{i \in \mathcal{V}} e_i + c$, $e_i$ represents the degree of agent $i$, and $c > 0$ is a constant.

In the following, we will show how NPGA is designed based on a variant of PUDA. Defining $\lambda = [\lambda^T_1, \ldots, \lambda^T_n]^T \in \mathbb{R}^{np}$, we can easily verify that (5) is equivalent to

$$\max_{\lambda \in \mathbb{R}^{np}} \phi(\lambda) - h^T(\lambda)$$

s.t. $\lambda_i = \lambda_j$, $i = 1, \ldots, n$ (6)

where $\phi(\lambda) = \sum_{i=1}^n \phi_i(\lambda_i)$ and $h^T(\lambda) = \frac{1}{n} \sum_{i=1}^n h_i^T(\lambda_i)$. Let $B \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{n \times n}$, and $D \in \mathbb{R}^{n \times n}$ be symmetric matrices associated with the network topology among agents. For (6), our variant of PUDA is implemented as

$$v^{k+1} = \lambda^k - C\lambda^k - By^k + \beta (\nabla \phi(\lambda^k) + \theta (\nabla \phi(\lambda^k) - \nabla \phi(\lambda^{k-1})))$$

$$y^{k+1} = y^k + \gamma Bv^{k+1}$$

$$\lambda^{k+1} = \text{prox}_{\alpha h^T} (Dv^{k+1})$$

where $B = B \otimes I_p$, $C = C \otimes I_p$, $D = D \otimes I_p$, $\beta$, $\gamma > 0$, and $\alpha > 0$.

**Remark 3:** The differences between (7) and the original PUDA are as follows.

1. Inspired by the PGA proposed in [34], we replace $\nabla \phi(\lambda^k)$ with $\nabla \phi(\lambda^k) + \theta (\nabla \phi(\lambda^k) - \nabla \phi(\lambda^{k-1}))$ in the primal update, which is quite useful to improve the actual performance of NPGA.
2. We introduce a step-size $\gamma$ in the dual update while in the original PUDA, it is always set to 1. In Section IV, we will see that the condition $\gamma < 1$ is crucial to guarantee the linear convergence of NPGA under weaker conditions.

Define

$$x^*_i(\lambda_i) = \arg\min_{x_i \in \mathbb{R}^{k_i}} \left\{ f_i(x_i) + g_i(x_i) + \lambda^T_i A_i x_i \right\}$$

when $f_i$ is strictly convex and $x^*_i(\lambda_i)$ is always unique, then $\phi_1$ is differentiable, and we have [36]

$$\nabla \phi_1(\lambda_i) = A_i x^*_i(\lambda_i).$$

As a result, $\phi$ is differentiable, and we have

$$\nabla \phi(\lambda) = [\nabla \phi_1(\lambda_1)^T, \ldots, \nabla \phi_n(\lambda_n)^T]^T = Ax^*(\lambda)$$

where $A = \text{diag}(A_1, \ldots, A_n)$. Let

$$x^*(\lambda) = [x^*_1(\lambda_1)^T, \ldots, x^*_n(\lambda_n)^T]^T = \arg\min_{x \in \mathbb{R}^k} \left\{ f(x) + g(x) + \lambda^T Ax \right\}$$

then (7) can be unfolded as

$$x^{*}(\lambda^k) = \arg\min_{x \in \mathbb{R}^k} \left\{ f(x) + g(x) + \lambda^T Ax \right\}$$

$$v^{k+1} = \lambda^k - C\lambda^k - By^k + \beta A \left( x^{*}(\lambda^k) + \theta (x^{*}(\lambda^k) - x^{*}(\lambda^{k-1})) \right)$$

$$y^{k+1} = y^k + \gamma Bv^{k+1}$$

$$\lambda^{k+1} = \text{prox}_{\beta h^T} (Dv^{k+1})$$

(8)

At each iteration of (8), we have to solve a subproblem exactly to obtain the dual gradient $Ax^*(\lambda^k)$. However, solving an optimization problem exactly is often costly and sometimes even impossible [37]. To address this issue, an inexact version of (8), i.e., NPGA, is further designed

$$x^{k+1} = \text{prox}_{\alpha g} (x^k - \alpha (\nabla f(x^k) + A^T \lambda^k))$$

$$v^{k+1} = \lambda^k - C\lambda^k - By^k + \beta A x^{k+1}$$

$$y^{k+1} = y^k + \gamma Bv^{k+1}$$

$$\lambda^{k+1} = \text{prox}_{\beta h^T} (Dv^{k+1})$$

$$\hat{x}^{k+1} = x^{k+1} + \theta (x^{k+1} - x^k)$$

(9)

where we replace the exact dual gradient with an inexact one updated by a step proximal gradient descent. Therefore, NPGA...
can be seen as an inexact version of (8). Alternatively, NPGA can also be seen as a decentralized version of PGA, where (9a) and (9b)–(9d) correspond to the primal and dual updates of PGA, respectively. Also note that (9b)–(9d) can be regraded as a special case of PGA; thus, NPGA includes two PGAs, and one is nested in the other one, which is why it is named NPGA.

**Remark 4:** As mentioned before, by choosing different $B$, $C$, and $D$, PUDA can recover many existing distributed unconstrained optimization algorithms. As an inexact dual version of PUDA, NPGA does inherit this property of PUDA, and we can also obtain many different versions of NPGA by choosing different $B$, $C$, and $D$, as shown in Table II, where the combinations of matrices for NPGA-I and NPGA-II are proposed in the experiment section of [17]. It is also worth noting that both DCDA and DCPA are special cases of NPGA: DCDA corresponds to NPGA-Exact diffusion with $\theta = 0$ and $\gamma = 1$ while DCPA corresponds to NPGA-P2D2 with $c = 1$ and $\theta = 1$. In numerical experiments, it has been shown that the convergence rates of different versions of NPGA usually have significant differences. This indicates that NPGA could provide an opportunity to design more efficient distributed algorithms for (P1).

**Remark 5:** To illustrate the decentralized implementation of NPGA in practice, we take NPGA-EXTRA as an example. Notice that NPGA can be rewritten as

\[
\begin{align*}
\mathbf{x}^{k+1} &= \operatorname{prox}_{\alpha g_i} \left( \mathbf{x}^k - \alpha (\nabla f_i(\mathbf{x}^k) + A_i^{\top} \lambda^k) \right), \\
\mathbf{v}^{k+1} &= (I - C)(\lambda^k - \lambda^{k-1}) + (I - \gamma B^2)\mathbf{v}^k + \beta A(\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k), \\
\lambda^{k+1} &= \operatorname{prox}_{\beta h_i} \left( D\mathbf{v}^{k+1} \right), \\
\hat{\mathbf{x}}^{k+1} &= \mathbf{x}^{k+1} + \theta(\mathbf{x}^{k+1} - \mathbf{x}^k)
\end{align*}
\]

where $W = W \otimes I_p$. Thus, NPGA-EXTRA is implemented as

\[
\begin{align*}
x_i^{k+1} &= \operatorname{prox}_{\alpha g_i} \left( x_i^k - \alpha(\nabla f_i(x_i^k) + A_i^{\top} \lambda_i^k) \right), \\
v_i^{k+1} &= \frac{1}{2}(\lambda_i^k - \lambda_i^{k-1}) + \frac{1}{2} \sum_{j=1}^{n} w_{ij} (\lambda_j^k - \lambda_j^{k-1} + \gamma v_j^k) + 2 - \gamma v_i^k + \beta A_i(\hat{x}_i - \hat{x}_i^k), \\
\lambda_i^{k+1} &= \operatorname{prox}_{\beta h_i} \left( v_i^{k+1} \right), \\
\hat{x}_i^{k+1} &= x_i^{k+1} + \theta(x_i^{k+1} - x_i^k)
\end{align*}
\]

for agent $i$, recall that $w_{ij} = 0$ if $i \neq j$ and agent $j$ is not the neighbor of agent $i$. Also note that agent $i$ only needs to send $\lambda_i^k - \lambda_i^{k-1} + \gamma v_i^k$ to its neighbors, hence NPGA-EXTRA requires only one round of communication per iteration. The decentralized implementations and the corresponding communication rounds of other versions of NPGA can be derived in a similar way.

### IV. Convergence Analysis

In this section, we analyze the linear convergence of NPGA under two cases: 1) $g_i = 0$ and $A$ has full row rank and 2) $h$ is $l_h$-smooth. Before proceeding with the analysis, we introduce a necessary assumption that applies to both cases.

**Assumption 4:** The network matrices $B$, $C$, and $D$ satisfy the following.

i) $C = 0$ or $\operatorname{Null}(C) = \operatorname{Span}(I_n)$.

ii) $\operatorname{Null}(B) = \operatorname{Span}(I_n)$.

iii) $0 < C < I, B^2 \leq I$.

iv) $D$ is a symmetric doubly stochastic matrix.

**Remark 6:** With the three properties of $W$ given in Remark 2, $\operatorname{Null}(L) = \operatorname{Span}(I_n)$, and $\operatorname{Null}(B) = \operatorname{Span}(I_n)$ is equivalent to $\operatorname{Null}(B^2) = \operatorname{Span}(I_n)$, we can easily verify that all algorithms listed in Table II satisfy (i), (ii), and (iv). Since $L \geq 0$, we can choose small enough $c$ and $\beta$ such that NPGA-DLM satisfies (iii). Note that $-I < \mathbf{W} \leq I$, then we have $0 \leq \mathbf{I} - \mathbf{W} < 2I$; hence, NPGA-EXTRA and NPGA-Exact diffusion both satisfy (iii), NPGA-P2D2 and NPGA-NIDS satisfy (iii) for $c \leq 1$ and $c \leq \frac{1}{2}$, respectively. Let $W' \triangleq \frac{1}{2}(I + W)$, obviously $0 < W' \leq I$. By replacing $W$ with $W'$, all the remaining algorithms can satisfy (iii).

**Table II**

**Different Versions of NPGA**

| Versions         | $B^2$       | $C$       | $D$       | Communication rounds |
|------------------|-------------|-----------|-----------|----------------------|
| NPGA-DIing       | $(I - W)^2$ | $I - W^2$ | $I$       | 2                    |
| NPGA-EXTRA       | $\frac{1}{2}(I - W')$ | $\frac{1}{2}(I - W)$ | $I$       | 1                    |
| NPGA-DLM         | $c_\beta L$ | $c_\beta L$ | $I$       | 1                    |
| NPGA-P2D2        | $\frac{1}{2}(I - W')$ | $\frac{1}{2}(I - W)$ | $I$       | 1                    |
| NPGA-Aug-DGM     | $(I - W)^2$ | 0         | $W^2$     | 2                    |
| NPGA-ATC tracking| $(I - W)^2$ | $I - W'$ | $W$       | 2                    |
| NPGA-Exact diffusion | $\frac{1}{2}(I - W')$ | 0         | $\frac{1}{2}(I - W')$ | 1         |
| NPGA-NIDS        | $c(I - W)$ | 0         | $I - c(I - W)$ | 2                    |
| NPGA-I           | $I - W$    | 0         | $W^2$     | 2                    |
| NPGA-II          | $I - W$    | $I - W$  | $W$       | 2                    |

The parameters used in this table correspond to those in Table I.
The following lemma establishes the relation between the fixed point of NPGA and the solution of the saddle-point problem (3).

**Lemma 3:** Assume Assumption 4 holds, \((x^*, \lambda^*)\) is a solution of the saddle-point problem (3) if and only if there exist \(v^* \in \mathbb{R}^{n_p}\) and \(y^* \in \mathbb{R}^{n_p}\) such that \((x^*, v^*, y^*, \gamma)\) is a fixed point of NPGA, where \(\gamma = 1_n \otimes \lambda^*\).

**Proof:** See the full version of this work [22].

The optimality condition of the saddle-point problem (3) is given by

\[
0 \in \nabla f(x^*) + A^\top \lambda^* + \partial g(x^*) \quad (11a)
\]
\[
A x^* \in \partial h^*(\lambda^*). \quad (11b)
\]

Then, we can verify that the uniqueness of the solution of (3) can be guaranteed by combining the strong convexity of \(f\) with the condition that \(A\) has full row rank or \(h^*\) is strongly convex, which implies that the solution of (3) is unique for both cases. Let \((x^*, v^*, y^*, \lambda^*)\) be a fixed point of (9), we can easily verify that \(x^*, v^*,\) and \(\lambda^*\) are unique, but \(y^*\) is not necessarily unique due to the singularity of \(B\). Nevertheless, \(y^*\) is, indeed, unique in \(\text{Col}(B)\), denoted as \(y^*_c\). If \(y^* = 0\), then we have \(y^k \in \text{Col}(B)\) for all \(k \geq 0\), which implies that, given the premise that NPGA converges, \(y^k\) can only converge to \(y^*_c\). Consequently, we only need to prove that NPGA linearly converges to \((x^*, v^*, y^*_c, \lambda^*)\) given the initial condition \(y^0 = 0\).

Define the error variables as

\[
\tilde{x}^k = x^k - x^*,
\]
\[
\tilde{v}^k = v^k - v^*,
\]
\[
\tilde{y}^k = y^k - y^*_c,
\]
\[
\tilde{\lambda}^k = \lambda^k - \lambda^*.
\]

Then, the error system of NPGA can be written as

\[
\tilde{x}^{k+1} = \tilde{x}^k - \alpha (\nabla f(x^k) - \nabla f(x^*)) - \alpha A^\top \tilde{\lambda}^k
\]
\[
- \alpha (S_g(x^{k+1}) - S_g(x^*))
\]
\[
\tilde{v}^{k+1} = \tilde{\lambda}^k - C \tilde{\lambda}^k - B \tilde{y}^k + \beta A (\tilde{x}^{k+1} + \theta(\tilde{x}^{k+1} - \tilde{x}^k))
\]
\[
\tilde{y}^{k+1} = \tilde{y}^k + \gamma B \tilde{v}^{k+1}
\]
\[
\tilde{\lambda}^{k+1} = \text{prox}_{\beta h^*} (D \tilde{v}^{k+1}) - \text{prox}_{\beta h^*} (D v^*).
\]

The following convergence analysis will be based on the aforementioned error system.

**A. Case I: \(g_i = 0\) and \(A\) Has Full Row Rank**

**Assumption 5:** \(g_i = 0\) and \(A\) has full row rank.

**Remark 7:** As mentioned before, Assumption 5 is the weakest known condition for decentralized algorithms to linearly solve (P3) with a nonsmooth \(h\). Before NPGA, there are two algorithms that can achieve linear convergence under the condition that \(A\) has full row rank: 1) DCPA, which is a special case of NPGA and 2) IDEA, which considers a special case of (P3), i.e., \(h\) is an indicator function corresponding to the equality constraint.

Since \(g_i = 0\), the error system (12) can be simplified to

\[
\tilde{x}^{k+1} = \tilde{x}^k - \alpha (\nabla f(x^k) - \nabla f(x^*)) - \alpha A^\top \tilde{\lambda}^k
\]
\[
\tilde{v}^{k+1} = \tilde{\lambda}^k - C \tilde{\lambda}^k - B \tilde{y}^k + \beta A (\tilde{x}^{k+1} + \theta(\tilde{x}^{k+1} - \tilde{x}^k))
\]
\[
\tilde{y}^{k+1} = \tilde{y}^k + \gamma B \tilde{v}^{k+1}
\]
\[
\tilde{\lambda}^{k+1} = \text{prox}_{\beta h^*} (D \tilde{v}^{k+1}) - \text{prox}_{\beta h^*} (D v^*).
\]

In order to establish the linear convergence of NPGA under the condition that \(A\) has full row rank, we need to introduce the following key lemma.

**Lemma 4:** Assume Assumption 5 holds, \(M \in \mathbb{R}^{n \times n}\) is a symmetric double stochastic matrix and \(H \in \mathbb{R}^{n \times n}\) is a positive semidefinite matrix, which satisfies \(\text{Null}(H) = \text{Span}(1_n)\), then \(\text{M}_A^\top M + \epsilon H > 0\), where \(M = M \otimes I_p\), \(H = H \otimes I_p\), and \(\epsilon \) is a constant.

**Proof:** See the full version of this work [22].

We now present an inequality that plays a crucial role in the subsequent analysis.

**Lemma 5:** Given Assumptions 1–5, let \(y^0 = 0, \theta \geq 0\), and the positive step-sizes satisfy

\[
\alpha < \frac{1}{l(1 + 2\theta)}, \quad \beta < \frac{\mu}{\pi^2(A)} \left( \frac{1}{1 - \sigma(C)} \right), \quad \gamma < 1
\]

then we have

\[
c_1 \|\tilde{x}^{k+1}\|^2 + c_2 \|\tilde{v}^{k+1}\|^2 \leq c_3 \|\tilde{y}^{k+1}\|^2
\]
\[
\leq \delta_1 c_1 \|\tilde{x}^k\|^2 + c_2 \|\tilde{\lambda}^k\|^2 \leq (0, 1), \quad \delta_3 = 1 - \gamma \sigma^2(B) \in (0, 1)
\]

**Proof:** See the full version of this work [22].

To obtain the linear convergence of NPGA, we need to impose another condition on \(C\), i.e., \(\text{Null}(C) = \text{Span}(1_n)\), which excludes the versions of NPGA with \(C = 0\).

**Assumption 6:** The network matrix \(B\) satisfies

\[
\text{Null}(C) = \text{Span}(1_n).
\]

Based on Lemma 5 and Assumption 6, we can obtain the following result.

**Lemma 6:** Under the same condition with Lemma 5, and assume Assumption 6 holds, then we have

\[
c_1 \|\tilde{x}^{k+1}\|^2 + c_2 \|\tilde{v}^{k+1}\|^2 \leq c_3 \|\tilde{y}^{k+1}\|^2
\]
\[
\leq \delta_1 c_1 \|\tilde{x}^k\|^2 + \delta_2 c_2 \|\tilde{\lambda}^k\|^2 + \delta_3 c_3 \|\tilde{y}^k\|^2
\]

where \(E = A A^\top + \frac{1}{2\alpha^2 \gamma}, C, \delta_2 = 1 - \alpha \beta \sigma^2(E) \in (0, 1)\), \(c_1, c_2, c_3, \delta_1\), and \(\delta_3\) are defined in Lemma 5.

**Proof:** See the full version of this work [22].
Now, we present the linear convergence result of NPGA under Case I.

**Theorem 1:** Given Assumptions 1–6, let $y^0 = 0, \theta \geq 0$, and the positive step-sizes satisfy

$$
\alpha < \frac{1}{l(1 + 2\theta)}, \quad \beta \leq \frac{\mu}{\sigma^2(A)} \left( \frac{1}{1 - \sigma(C)} + \theta \right),
$$

$$
\gamma < \min \left\{ 1, \frac{\alpha \beta \theta (E)}{\sigma^2(B)} \right\}
$$

(17)

then there exists $\delta \in (0, 1)$ such that

$$
\|x^k - x^\star\|^2 = O(\delta^k)
$$

(18)

where

$$
\delta = \max \left\{ 1 - \alpha \mu (1 - \alpha l(1 + 2\theta)), \frac{1 - \alpha \beta \theta (E)}{1 - \gamma \sigma^2(B)}, \frac{1}{\gamma} - \gamma^2(B) \right\}.
$$

**Proof:** Recall that $D$ is a symmetric doubly stochastic matrix, which implies that $\|D\| < 1$, then applying the nonexpansive property of the proximal operator gives that

$$
\|\tilde{x}^{k+1}\|^2 = \|\text{prox}_{\beta h_r} (Dv^{k+1}) - \text{prox}_{\beta h_r} (Dv^\star)\|^2
$$

$$
\leq \|Dv^{k+1}\|^2
$$

$$
\leq \|\tilde{y}^{k+1}\|^2
$$

Note that the condition of Theorem 1 is sufficient for Lemma 6, we have

$$
c_1 \|x^{k+1}\|^2 + \left( 1 - \gamma \sigma^2(B) \right) c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq c_1 \|x^{k+1}\|^2 + \left( 1 - \gamma \sigma^2(B) \right) c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq c_1 \|x^{k+1}\|^2 + c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq \delta_1 c_1 \|x^{k}\|^2 + \delta_2 c_2 \|\tilde{x}^{k}\|^2 + \delta_3 c_3 \|y^{k}\|^2.
$$

(19)

Let $\omega = 1 - \gamma \sigma^2(B)$, obviously $\omega \in (0, 1)$, we then have

$$
c_1 \|x^{k+1}\|^2 + \left( 1 - \gamma \sigma^2(B) \right) c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq \delta_1 c_1 \|x^{k}\|^2 + \delta_2 c_2 \|\tilde{x}^{k}\|^2 + \delta_3 c_3 \|y^{k}\|^2
$$

$$
\leq \delta \left( c_1 \|x^{k}\|^2 + c_2 \|\tilde{x}^{k}\|^2 + c_3 \|y^{k}\|^2 \right)
$$

(20)

where $\delta = \max(\delta_1, \frac{\delta_2}{\omega}, \delta_3)$. It follows that

$$
c_1 \|x^{k}\|^2 \leq \delta^k \left( c_1 \|x^0\|^2 + c_2 \|\tilde{x}^0\|^2 + c_3 \|y^0\|^2 \right)
$$

(21)

the step-sizes condition (14) implies that $\frac{\delta_2}{\omega} \in (0, 1)$; thus, $\delta \in (0, 1)$.

**Remark 8:** Given a sequence $\{x^k\}$ that converges to $x^\star$, the convergence is called 1) R-linear if there exists $\lambda \in (0, 1)$ such that $\|x^k - x^\star\| \leq C\lambda^k$ for $k \geq 0$, where $C$ is a positive constant; 2) Q-linear if there exists $\lambda \in (0, 1)$ such that $\frac{\|x^{k+1} - x^\star\|}{\|x^k - x^\star\|} \leq \lambda$ for $k \geq 0$ [24]. Therefore, (18) implies that $x^k$ converges to $x^\star$ with an R-linear rate of $\sqrt{\delta}$.

If the following assumption is further assumed, we can obtain a tighter convergence rate than Theorem 1.

**Assumption 7:** The network matrices $B$ and $D$ satisfy

$$
D^2 \leq I - B^2.
$$

**Remark 9:** We can easily see that all CTA versions of NPGA (where $D = I$) cannot satisfy Assumption 7, only ATC ones have the possibility. Since Theorem 2 requires both Assumptions 6 and 7, we only analyze ATC versions of NPGA with $C \neq 0$, i.e., NPGA-ATC tracking and NPGA-II. Recall that $W \preceq I$, which implies that $W^2 \preceq W$, then we can easily verify that NPGA-ATC tracking and NPGA-II both satisfy Assumption 7 according to Table II.

**Theorem 2:** Given Assumptions 1–7, let $y^0 = 0, \theta \geq 0$, and the positive step-sizes satisfy

$$
\alpha < \frac{1}{l(1 + 2\theta)}, \quad \beta \leq \frac{\mu}{\sigma^2(A)} \left( \frac{1}{1 - \sigma(C)} + \theta \right), \quad \gamma < 1
$$

(22)

then there exists $\delta \in (0, 1)$ such that

$$
\|x^k - x^\star\|^2 = O(\delta^k)
$$

(18)

where

$$
\delta = \max \left\{ 1 - \alpha \mu (1 - \alpha l(1 + 2\theta)), 1 - \alpha \beta \theta (E), 1 - \gamma \sigma^2(B) \right\}.
$$

**Proof:** According to Assumption 7, we have

$$
D^2 \leq I - B^2 \leq I - \gamma B^2
$$

it follows that:

$$
\|\tilde{x}^{k+1}\|^2 = \|\text{prox}_{\beta h_r} (Dv^{k+1}) - \text{prox}_{\beta h_r} (Dv^\star)\|^2
$$

$$
\leq \|Dv^{k+1}\|^2
$$

$$
\leq \|\tilde{y}^{k+1}\|^2
$$

Note that the condition of Theorem 2 is sufficient for Lemma 6, we have

$$
c_1 \|x^{k+1}\|^2 + \left( 1 - \gamma \sigma^2(B) \right) c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq c_1 \|x^{k+1}\|^2 + \left( 1 - \gamma \sigma^2(B) \right) c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq c_1 \|x^{k+1}\|^2 + c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq \delta_1 c_1 \|x^{k}\|^2 + \delta_2 c_2 \|\tilde{x}^{k}\|^2 + \delta_3 c_3 \|y^{k}\|^2
$$

(19)

Let $\omega = 1 - \gamma \sigma^2(B)$, obviously $\omega \in (0, 1)$, we then have

$$
c_1 \|x^{k+1}\|^2 + \left( 1 - \gamma \sigma^2(B) \right) c_2 \|\tilde{x}^{k+1}\|^2 + c_3 \|y^{k+1}\|^2
$$

$$
\leq \delta_1 c_1 \|x^{k}\|^2 + \delta_2 c_2 \|\tilde{x}^{k}\|^2 + \delta_3 c_3 \|y^{k}\|^2
$$

$$
\leq \delta \left( c_1 \|x^{k}\|^2 + c_2 \|\tilde{x}^{k}\|^2 + c_3 \|y^{k}\|^2 \right)
$$

(20)

where $\delta = \max(\delta_1, \frac{\delta_2}{\omega}, \delta_3)$. It follows that

$$
c_1 \|x^{k}\|^2 \leq \delta^k \left( c_1 \|x^0\|^2 + c_2 \|\tilde{x}^0\|^2 + c_3 \|y^0\|^2 \right)
$$

(21)

the step-sizes condition (14) implies that $\frac{\delta_2}{\omega} \in (0, 1)$; thus, $\delta \in (0, 1)$.

**Remark 10:** Compared to Theorem 1, the improved convergence result in this case relaxes the upper bound on $\gamma$ to 1 and provides a tighter convergence rate of $\max(1 - \alpha \mu (1 - \alpha l(1 + 2\theta)),$ $1 - \alpha \beta \theta (E),$ $1 - \gamma \sigma^2(B))$.

□
all(1 + 2θ), 1 − αβη(E), 1 − αγ(B). It is worth noting that the convergence of DCPA under Case I can be encompassed by Theorem 1 with negligible differences. Therefore, we can conclude that the ATC versions of NPGA, which satisfy Assumption 7, achieve tighter convergence rates than DCPA.

Theorems 1 and 2 both assume Assumption 6 holds, which cannot be satisfied by the versions of NPGA with C = 0. Fortunately, we find that the linear convergence of NPGA can be guaranteed without Assumption 6, when h is an indicator function that corresponds to the equality constraint.

**Assumption 8:** h is an indicator function given as
\[ h(y) = \begin{cases} 0, & y = b \\ ∞, & y \neq b \end{cases} \]
where \( b = \frac{1}{n}1_n \otimes b \); then, the error system (13) can be simplified to
\[ \dot{x}^{k+1} = x^k - \alpha (\nabla f(x^k) - \nabla f(x^k)) - \alpha A^T \lambda^k \]
\[ \dot{v}^{k+1} = \dot{\lambda}^k - C\lambda^k - B\ddot{v}^k + \beta A (x^{k+1} + \theta(x^{k+1} - x^k)) \]
\[ \dot{y}^{k+1} = y^k + \gamma B\dot{v}^{k+1} \]
\[ \dot{\lambda}^{k+1} = D\dot{v}^{k+1}. \]  
(24)

Different from Theorems 1 and 2, we replace Assumption 6 with the following much weaker assumption, which can be satisfied by all versions of NPGA listed in Table II.

**Assumption 9:** The network matrices B, C, and D satisfy
\[ D(I - C)D \leq I - B^2. \]

**Remark 11:** We analyze Assumption 9 for the ATC and CTA versions of NPGA separately.

1) For CTA versions (where \( D = I \)), Assumption 9 is simplified to \( B^2 \leq C \). Obviously, NPGA-EXTRA and NPGA-DLM satisfy this condition, and NPGA-P2D2 can satisfy it by choosing \( c \leq 1 \). Recall that \( W \leq I \), which implies that \( W^2 \leq W \); hence, DIG also satisfies Assumption 9.

2) By utilizing the properties \( W \leq I \) and \( W^2 \leq W \), we can easily verify that all the ATC versions of NPGA listed in Table II satisfy Assumption 9, where NPGA-NDS needs to choose \( c \leq \frac{1}{2} \) to satisfy the condition.

**Theorem 3:** Given Assumptions 1–5, 8, 9, let \( y^0 = 0, \theta = 0 \), and the positive step-sizes satisfy
\[ \alpha < \frac{1}{\gamma}, \quad \beta \leq \frac{\mu(1 - \sigma(C))}{\sigma^2(A)}, \quad \gamma < 1 \]  
then there exists \( \delta \in (0, 1) \) such that
\[ \|x^k - x^*\|^2 = O(\delta^k) \]
where \( \delta = \max \{1 - \alpha(1 - \alpha), 1 - \alpha\beta\eta(F), 1 - \alpha^2\} \)
\[ F = DAA^TD + \frac{1 - \gamma}{\alpha\beta}B^2. \]

**Proof:** Note that \( \text{Null}(B) = \text{Span}(1_n) \) is equivalent to \( \text{Null}(B^2) = \text{Span}(1_n) \) and recall that D is double stochastic, we can verify that \( F > 0 \) by Lemma 4; then, we have
\[ \|\dot{x}^k\|^2_{I - \alpha\beta} \leq (1 - \alpha\beta\eta(F))\|\dot{x}^k\|^2. \]  
(26)

Also note that \( \sigma(D) \leq 1 \), according to Lemma 2, we have
\[ \sigma(A^TD) \leq \sigma(A^T)\sigma(D) \leq \sigma(A). \]
Let \( \delta_2 = 1 - \alpha\beta\eta(F) \), applying Weyl’s inequality [38] gives that
\[ \frac{\eta(F)}{\sigma^2(A^T)} = \eta_1(F) \]
\[ \leq \eta\left(1 - \frac{\sigma^2(A^T)}{\sigma^2(A)}\right) \]
\[ \leq \sigma^2(A) \]  
(27)
where the first inequality holds since \( \eta_1(B^2) = 0 \). According to (25) and \( \mu \leq 1 \), we have
\[ \alpha\beta \leq \frac{1 - \sigma(C)}{\sigma^2(A)} \]
combining it with (27) gives that \( \delta_2 \in (0, 1) \). Note that the condition of Theorem 3 is sufficient for Lemma 5, then we have
\[ c_1\|x^k\|^2 + c_2\|v^k\|^2_{I - \gamma B^2} + c_3\|y^k\|^2_{I - \gamma} \]
\[ \leq \delta_1 c_1\|x^k\|^2 + c_2\|v^k\|^2_{D(I - C - \alpha\beta\AA^T)} + c_3\|y^k\|^2_{I - \gamma} \]
\[ \leq \delta_1 c_1\|x^k\|^2 + c_2\|v^k\|^2_{D(I - C - \alpha\beta\AA^T)} + \delta_3 c_3\|y^k\|^2 \]
\[ \leq \delta_1 c_1\|x^k\|^2 + c_2\|v^k\|^2_{D(I - C - \alpha\beta\AA^T)} + \delta_3 c_3\|y^k\|^2 \]
\[ \leq \delta_1 c_1\|x^k\|^2 + c_2\|v^k\|^2_{D(I - C - \alpha\beta\AA^T)} + \delta_3 c_3\|y^k\|^2 \]
\[ \leq \delta_1 c_1\|x^k\|^2 + c_2\|v^k\|^2_{D(I - C - \alpha\beta\AA^T)} + \delta_3 c_3\|y^k\|^2 \]
\[ \leq \delta_1 c_1\|x^k\|^2 + c_2\|v^k\|^2_{D(I - C - \alpha\beta\AA^T)} + \delta_3 c_3\|y^k\|^2 \]
\[ \leq \delta \left(c_1\|x^k\|^2 + c_2\|v^k\|^2_{I - \gamma B^2} + c_3\|y^k\|^2_{I - \gamma}\right) \]
(28)
where the second, the third, the fifth, and the sixth inequalities hold due to (24), Assumption 9, (26), and \( \delta_2 < 1 \), respectively. It follows that
\[ c_1\|x^k\|^2 \leq \delta^k \left(c_1\|x^0\|^2 + c_2\|v^0\|^2_{I - \gamma B^2} + c_3\|y^0\|^2_{I - \gamma}\right). \]

**Remark 12:** As shown in Theorem 3, when Assumption 8 holds, the linear convergence of NPGA can be guaranteed without assuming \( \text{Null}(C) = \text{Span}(1_n) \). The possibility of \( C = 0 \) introduces an interesting outcome: The upper bounds of the step sizes of NPGA are independent of the network topology when \( C = 0 \), which expands their stability region, surpassing other versions of NPGA, including DCPA. Furthermore, the upper bound of \( \gamma \) and the linear convergence rate are both improved.
compared to Theorem 1. Recall that DCPA is a special case of NPGA; hence, this improvement can also be applied to it. Also note that most of the ATC versions of NPGA are with $C = 0$, which are excluded in Theorem 1. However, as shown in numerical experiments, ATC algorithms usually have much better performance than CTA ones. Therefore, the improvement of Theorem 3 is highly significant and meaningful.

**B. Case II: $h$ is $l_h$-Smooth**

**Assumption 10:** $h$ is $l_h$-smooth.

Assumption 10 implies that $h^*$ is $\frac{1}{l_h}$-strongly convex; then, we can obtain the linear convergence of NPGA without imposing the full rank condition on $A$, as shown in the following theorem.

**Theorem 4:** Given Assumptions 1–4, 10, let $\mathbf{x}^0 = 0, \theta \geq 0$, and the positive step-sizes satisfy

\[
\alpha < \min \left\{ \frac{\mu}{l(2 + 2\theta)}, \frac{1}{2l - \mu} \right\}, \quad \beta \leq \frac{\mu}{\pi^2(A) \left( \frac{1}{1 - \pi(C)} + \theta \right)}
\]

\[
\gamma < \min \left\{ \frac{1}{\left( 1 + \frac{\beta}{\tau} \right)^2 - 1}, \frac{\left( 1 + \frac{\beta}{\tau} \right)^2 - 1}{\left( 1 + \frac{\beta}{\tau} \right)^2 \pi^2(B)} \right\}
\]

then there exists $\delta \in (0, 1)$ such that

\[
\| \mathbf{x}^k - \mathbf{x}^* \|^2 = O \left( \delta^k \right)
\]

where

\[
\delta = \max \left\{ 1 - \alpha \left( \mu - 2\theta \alpha^2 \right), \frac{1}{\left( 1 + \frac{\beta}{\tau} \right)^2 \left( 1 - \gamma \pi^2(B) \right)}, 1 - \gamma \sigma^2(B) \right\}.
\]

**Proof:** See the full version of this work [22].

**V. NUMERICAL EXPERIMENTS**

In this section, we validate NPGAs convergence and evaluate the performance by solving vertical federated learning problems. Notice that NPGA-EXTRA and NPGA-Exact diffusion are the special cases of NPGA-P2D2 and NPGA-NIDS, respectively; hence, we exclude them from the experiments. Although DCPA is also a special case of NPGA, we include it for the purpose of comparison.

Consider a scenario with $n$ parties, where each party has its own private data that cannot be accessed by others. The goal of federated learning systems is to train a learning model using all parties’ data without compromising individual privacy. In vertical federated learning, the datasets among different parties consist of the same samples but with different features [3].

In particular, we focus on vertical federated learning for three types of regression problems: 1) ridge regression, 2) logistic regression, and 3) elastic net regression, which correspond to the settings of Theorems 1, 3, and 4, respectively. For a given regression problem, let $X = \{X_1, \ldots, X_n\} \in \mathbb{R}^{p \times d}$ and $Y \in \mathbb{R}^p$ be the feature matrix and the label vector, respectively. Here,

**A. Experiment I: Ridge Regression**

One form of the ridge regression problem is formulated as [12]

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{2} \| \theta \|^2 + h(X\theta)
\]

where $h$ is an indicator function given as

\[
h(x) = \begin{cases}
0, & \|x - Y\| \leq \delta \\
\infty, & \|x - Y\| > \delta.
\end{cases}
\]

We use $X\theta$ to denote the linear model since the last element of $\theta$ is the intercept. Therefore, $X$ is defined by $X = [X_{\text{raw}}, I_n]$, where $X_{\text{raw}}$ is the raw feature matrix. Considering the vertical federated learning setting, (29) can be reformulated as

\[
\min_{\theta_i \in \mathbb{R}^d_i} \frac{1}{2} \sum_{i=1}^n \| \theta_i \|^2 + h \left( \sum_{i=1}^n X_i \theta_i \right)
\]

which can be covered by (P1) with $f_i(x_i) = \frac{1}{2} \|x_i\|^2, q_i(x_i) = 0$, and $A_i = X_i$. If $X$ has full row rank, (30) will match the setting of Theorem 1.

For this experiment, we use the Boston housing prices dataset, which has 13 features. We choose 10 samples to parties, where each party has its own private data that cannot be accessed by others. The goal of federated learning systems is to train a learning model using all parties’ data without compromising individual privacy. In vertical federated learning, the datasets among different parties consist of the same samples but with different features [3].

In particular, we focus on vertical federated learning for three types of regression problems: 1) ridge regression, 2) logistic regression, and 3) elastic net regression, which correspond to the settings of Theorems 1, 3, and 4, respectively. For a given regression problem, let $X = \{X_1, \ldots, X_n\} \in \mathbb{R}^{p \times d}$ and $Y \in \mathbb{R}^p$ be the feature matrix and the label vector, respectively. Here,

**B. Experiment II: Logistic Regression**

Given $X = [x_1, \ldots, x_p] \in \mathbb{R}^{p \times d}$ and $Y = [y_1, \ldots, y_p] \in \{1, -1\}^p$, the logistic regression problem is defined as

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{p} \sum_{i=1}^p \ln \left( 1 + \exp(-y_i \theta^\top x_i) \right) + \frac{\rho}{2} \| \theta \|^2
\]

where $\rho > 0$ is the regularization parameter. Considering the vertical federated learning setting, let $X_i = [x_{i1}, \ldots, x_{ip}]$, (31) can be rewritten as

\[
\min_{\theta_i \in \mathbb{R}^d_i} \frac{1}{p} \sum_{j=1}^p \ln \left( 1 + \exp \left( -y_j \sum_{i=1}^n \theta_i^\top x_{ij} \right) \right) + \frac{\rho}{2} \sum_{i=1}^n \| \theta_i \|^2.
\]

By introducing a slack variable $z = [z_1, \ldots, z_p] = \sum_{i=1}^{n-1} X_i \theta_i$, we can reformulate (32) as

\[
\min_{\theta_i \in \mathbb{R}^d_i, z \in \mathbb{R}^p} \frac{1}{p} \sum_{j=1}^p \ln \left( 1 + \exp \left( -y_j z_j \right) \right) + \frac{\rho}{2} \sum_{i=1}^n \| \theta_i \|^2
\]

s.t. $\sum_{i=1}^n X_i \theta_i - z = 0$

---

[2]Online. Available: http://lib.stat.cmu.edu/datasets/boston
which can be covered by (P1) with

\[ f_i(x_i) = \frac{\rho}{2} \|x_i\|^2 \quad \forall i \leq n \]

\[ f_n(x_{n+1}) = \frac{1}{p} \sum_{j=1}^{p} \ln (1 + \exp (-y_j x_{nj})) \]

\[ A_i = X_i \quad \forall i \leq n, \quad A_{n+1} = -I_p \]

A fact of (33) is that \( A = [X_1, \ldots, -I_p] \) always has full row rank, which enables it to match the setting of Theorem 3 naturally.
For this experiment, we use the Covtype dataset, which has 54 features. We choose 100 samples to construct $X$ and divide it into $[X_1, \ldots, X_{27}]$, where $X_i \in \mathbb{R}^{100 \times 2}$ for $i \leq 26$, and $X_{27} \in \mathbb{R}^{100 \times 3}$.

### C. Experiment III: Elastic Net Regression

The elastic net regression problem is formulated as

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2n} \|X\theta - Y\|^2 + \alpha \rho \|\theta\|_1 + \frac{\alpha(1-\rho)}{2} \|\theta\|^2$$

which is equivalent to

$$\min_{\theta \in \mathbb{R}^d} h(X\theta) + \alpha \rho \|\theta\|_1 + \frac{\alpha(1-\rho)}{2} \|\theta\|^2$$

where $h(x) = \frac{1}{2n} \|x - Y\|^2$. Considering the vertical federated learning setting, (34) can be reformulated as

$$\min_{\theta_i \in \mathbb{R}^{d_i}} \frac{\alpha(1-\rho)}{2} \sum_{i=1}^{n} \|\theta_i\|^2 + \alpha \rho \sum_{i=1}^{n} \|\theta_i\|_1 + h\left(\sum_{i=1}^{n} X_i \theta_i\right)$$

which can be covered by (P1) with $f_i(x_i) = \frac{\alpha(1-\rho)}{n} \|x_i\|^2$, $g_i(x_i) = \alpha \rho \|x_i\|_1$, and $A_i = X_i$. Note that $h$ is $\frac{1}{n}$-smooth; hence, (36) matches the setting of Theorem 4.

For this experiment, we still use the Boston housing prices dataset and choose ten samples to construct $X$. The division mode of $X$ is also the same as Experiment I. The only difference is that we do not need to guarantee that $X$ has full row rank.

According to the problem formulations of the previous three experiments, the number of parties is 13, 28, and 13, respectively. We utilize the Erdos–Renyi model [39] with a connectivity probability of 0.3 to generate the network topology among different parties, and the mixing matrix $W$ is constructed by the Laplacian method. When applying an algorithm to solve the problem, we try different combinations of step sizes to obtain the fastest convergence rate.

The experimental results are shown in Figs. 1–3, where the optimality gap is defined as $\frac{\|x^* - x\|}{\|x^* - x\|}$. Note that in various versions of NPGA, including DCPA, the computations of the gradient and the proximal operator represent the primary computational burden at each iteration. Furthermore, these computations are performed only once per iteration for both NPGA and DCPA. Consequently, the number of iterations can serve as a metric for comparing the computational costs between NPGA and DCPA. According to the experiment results, we can see that NPGA can achieve linear convergence for all problem settings, which supports our theoretical results. Besides, certain versions of NPGA exhibit significantly faster convergence rates compared to DCPA in terms of both the number of iterations and communication rounds, which suggests that they outperform DCPA in terms of both computational and communication efficiencies.

### VI. Conclusion

In this article, we study a class of decentralized constraint-coupled optimization problems and propose a novel nested primal–dual gradient algorithm called NPGA. NPGA not only serves as an algorithm but also offers an algorithmic framework, encompassing various existing algorithms as special cases. We prove that NPGA can achieve linear convergence under the weakest known condition, and the theoretical convergence rate surpasses all known results. By designing different network matrices, we can derive various versions of NPGA and analyze their convergences conveniently, providing an opportunity to design more efficient algorithms. Numerical experiments also confirm that NPGA exhibits a faster convergence rate compared to existing algorithms. Nevertheless, the linear convergence of NPGA is established under the assumption of a time-invariant network topology, which may not hold true in certain scenarios. Therefore, exploring the extension of NGPA linear convergence to time-varying graphs will be a meaningful direction of future research.

### References

[1] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, “Tracking-ADMM for distributed constraint-coupled optimization,” Automatica, vol. 117, 2020, Art. no. 108962.

[2] T.-H. Chang, M. Hong, and X. Wang, “Multi-agent distributed optimization via inexact consensus ADMM,” IEEE Trans. Signal Process., vol. 63, no. 2, pp. 482–497, Jan. 2014.

[3] Y. Liu et al., “Vertical federated learning,” 2022, arXiv:2211.12814.

[4] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Peschel, “Distributed basis pursuit,” IEEE Trans. Signal Process., vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[5] P. Yi, Y. Hong, and F. Liu, “Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems,” Automatica, vol. 74, pp. 259–269, 2016.

[6] Y. Zhu, W. Ren, W. Yu, and G. Wen, “Distributed resource allocation over directed graphs via continuous-time algorithms,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 51, no. 2, pp. 1097–1106, Feb. 2021.

[7] J. Zhang, K. You, and K. Cai, “Distributed dual gradient tracking for resource allocation in unbalanced networks,” IEEE Trans. Signal Process., vol. 68, pp. 2186–2198, 2020.

[8] S. S. Kia, “Distributed optimal in-network resource allocation algorithm design via a control theoretic approach,” Syst. Control Lett., vol. 107, pp. 49–57, 2017.

[9] A. Nedić, A. Olshevsky, and W. Shi, “Improved convergence rates for distributed resource allocation,” in Proc. IEEE Conf. Decis. Control, 2018, pp. 172–177.

[10] S. A. Alghunaim, K. Yuan, and A. H. Sayed, “A proximal diffusion strategy for multi-agent optimization under uncoordinated constant step-sizes,” IEEE Trans. Autom. Control, vol. 65, no. 11, pp. 4554–4567, Nov. 2020.

[11] J. Li and H. Su, “Implicit tracking-based distributed constraint-coupled optimization,” IEEE Trans. Control Netw. Syst., vol. 10, no. 1, pp. 479–490, Mar. 2023.

[12] S. A. Alghunaim, Q. Lyu, M. Yan, and A. H. Sayed, “Dual consensus proximal algorithm for multi-agent sharing problems,” IEEE Trans. Signal Process., vol. 69, pp. 5568–5579, 2021.

[13] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, “Geometrically convergent distributed optimization with uncoordinated step-sizes,” in Proc. Amer. Control Conf., 2017, pp. 3950–3955.

[14] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, “Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant step-sizes,” in Proc. IEEE 54th Conf. Decis. Control, 2015, pp. 2055–2060.

[15] Z. Li, W. Shi, and M. Yan, “A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates,” IEEE Trans. Signal Process., vol. 67, no. 17, pp. 4494–4506, Sep. 2019.

---

3Online. [Available]: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
[16] A. H. Sayed, “Diffusion adaptation over networks,” in *Academic Press Library in Signal Processing*, vol. 3. Amsterdam, The Netherlands: Elsevier, 2014, pp. 323–453.

[17] S. A. Alghunaim, E. K. Ryu, K. Yuan, and A. H. Sayed, “Decentralized proximal gradient algorithms with linear convergence rates,” *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2787–2794, Jun. 2021.

[18] J. Xu, Y. Tian, Y. Sun, and G. Scutari, “Distributed algorithms for composite optimization: Unified framework and convergence analysis,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3555–3570, 2021.

[19] J. Li and H. Su, “Gradient tracking: A unified approach to smooth distributed optimization,” 2022, arXiv:2202.09804.

[20] X. Wu and J. Lu, “A unifying approximate method of multipliers for distributed composite optimization,” *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2154–2169, Apr. 2023.

[21] Y. Nesterov, *Lectures on Convex Optimization*, vol. 137. Berlin, Germany: Springer, 2018.

[22] J. Li and H. Su, “NPGA: A unified algorithmic framework for decentralized constraint-coupled optimization,” 2022, arXiv:2205.11119.

[23] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[24] A. Nedić, A. Olshevsky, and W. Shi, “Achieving geometric convergence for distributed optimization over time-varying graphs,” *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[25] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[26] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, “DLM: Decentralized linearized alternating direction method of multipliers,” *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4051–4064, Aug. 2015.

[27] S. A. Alghunaim, K. Yuan, and A. H. Sayed, “A linearly convergent proximal gradient algorithm for decentralized optimization,” in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 2848–2858.

[28] P. D. Lorenzo and G. Scutari, “Next: In-network nonconvex optimization,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[29] G. Scutari and Y. Sun, “Distributed nonconvex constrained optimization over time-varying digraphs,” *Math. Program.*, vol. 176, no. 1, pp. 497–544, 2019.

[30] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning—Part I: Algorithm development,” *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 708–723, Feb. 2019.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[32] J. Li and H. Su, “Distributed nonconvex constrained optimization over time-varying digraphs,” *Math. Program.*, vol. 176, no. 1, pp. 497–544, 2019.

[33] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning—Part I: Algorithm development,” *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 708–723, Feb. 2019.

[34] G. Scutari and Y. Sun, “Distributed nonconvex constrained optimization over time-varying digraphs,” *Math. Program.*, vol. 176, no. 3, pp. 708–723, Feb. 2019.

[35] X. Gu, C. Wu, and H. Su, “Distributed optimization and learning—Part I: Algorithm development,” *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 708–723, Feb. 2019.

[36] D. P. Bertsekas, *Nonlinear Programming*. Nashua, NH, USA: Athena Scientific, 1999.

[37] O. Devolder, F. Glineur, and Y. Nesterov, “First-order methods of smooth convex optimization with inexact oracle,” *Math. Program.*, vol. 146, no. 1, pp. 37–75, 2014.

[38] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 2012.

[39] P. Erdős and A. Rényi, “On the evolution of random graphs,” *Publication Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.