SCIENCE CHINA
Information Sciences

• Supplementary File •

Distributed optimization algorithm with superlinear convergence rate

Yeming XU[1], Ziyuan GUO[1], Kaihong LU[1], Hongxia WANG[1] & Huanshui ZHANG[1,2*]

[1]*School of Electrical and Automation Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*School of Control Science and Engineering, Shandong University, Jinan 250061, China*

Appendix A Literature review

The distributed optimization problems are not only required to achieve consensus but also to cooperatively minimize a global objective function, where the global objective function is the sum of local objective functions, and each local objective function is only known to the agent itself. It has been widely studied in recent years for its broad applicability to sensor networks [1–3], distributed control [4–6], and machine learning [7–9].

In the context of distributed optimization, common approaches for tackling problem (1) include decentralized gradient descent (DGD) [10–12], the distributed versions of the alternating direction method of multipliers [13–15], and the decentralized dual averaging (DDA) algorithm [16]. These algorithms are characterized by their simplicity and low computational complexity. However, relying solely on first-order information is disadvantageous for the convergence rate. The fastest convergence rate attainable by distributed optimization algorithms that employ first-order methods is known to be linear [17].

Due to the superior convergence rate of second-order methods, they are regarded as appealing candidates for distributed implementation. Although the Hessian matrix shares the same sparsity pattern as the network, its inverse is typically dense, which hinders the straightforward application of second-order methods in distributed optimization. For example, the OCP method proposed in [18] is a novel optimization method based on optimal control, and it has already been successfully applied in trajectory tracking [19], yet its dependence on the Hessian matrix inverse hinders distributed implementation. To overcome this difficulty, researchers have introduced various Newton-type methods, leading to remarkable advancements in recent years. In [20], the authors proposed the distributed Adaptive Newton (DAN) method, achieving quadratic convergence but relying on a finite-time consensus inner loop in each step. Moreover, in a master-slave network configuration, prior works [21–23] proposed some distributed quasi-Newton methods where the master node processes complete information from all slaves. The network Newton (NN) algorithm achieves an approximate Newton step by truncating the Taylor series of the Hessian matrix inverse [24]. The distributed quasi-Newton (DQN) method belongs to a distinct family of Newton-type methods designed to solve problem (1) in a distributed manner [25]. Owing to the penalty reformulation, the convergence of DQN and NN is limited to a neighborhood around the solution. Qu et al. developed two second-order methods with an adapt-then-combine strategy, but they require the inversion of the local Hessian and converge only linearly to a neighborhood around the solution [26]. Second-order methods in the primal-dual domain are proposed in [14, 27–29], with the objective of improving both the convergence rate and the accuracy of the obtained solutions. The distributed Newton-Raphson method in [30] estimates the global Newton direction through the exchange of gradient and Hessian matrix information. The Newton tracking algorithm updates local variables using a Newton direction refined with neighbor and history data, yet still demands the inversion of the regularized local Hessian at each step, which increases the computational cost of the algorithm [31]. Both the Newton-Raphson and Newton tracking algorithms converge linearly to the optimal solution. To alleviate the computational cost, Ye et al. extended the inexact Newton method to propose a new decentralized second-order algorithm, but the convergence rate remains linear [32]. The distributed inexact Newton method with adaptive step size (DINAS) can operate without any local Hessian inverse calculations and demonstrate significant improvements of DINAS over existing alternatives [33]. The dual inexact nonsmooth Newton method based on Lagrangian dual decomposition guarantees superlinear convergence [34]. However, it requires solving an internal optimization problem at each iteration, which also leads to a higher computational burden. Although these methods successfully incorporate second-order information into distributed optimization, they primarily focus on directly modifying or truncating the Newton step. Such an approach is limited by the structure of Newton's method, which may affect the convergence rate of algorithms. To overcome this limitation, we need more flexible methods to utilize second-order information.

Appendix B Notations and remarks

Throughout the paper, the superscript $\top$ stands for matrix or vector transposition; $\|\cdot\|$ denotes the l_2-norm for vectors and the spectral norm for matrices; $\mathbb{R}^n$ is the set of n-dimensional real vectors; $\mathbb{S}^n_{++}$ denotes the set of real symmetric positive definite matrices; We use I_n and $\mathbf{0}_n$ to denote the $n \times n$ identity matrix and the n-dimensional zero vector, respectively; For vectors $v_i, i = 1, \ldots, n$, we use $\mathrm{col}\{v_i\}_{i=1}^n$ or $\mathrm{col}\{v_1, \ldots, v_n\}$ to denote the column vector obtained by stacking $v_1, \ldots, v_n$ vertically. For matrices $A_i, i = 1, \ldots, n$, $\mathrm{diag}\{A_1, \ldots, A_n\}$ denotes the block-diagonal matrix whose diagonal blocks are $A_1, \ldots, A_n$. For symmetric matrices A and B, $A \succ B$

* Corresponding author (email: hszhang@sdu.edu.cn)

means the matrix $A-B$ is positive semidefinite. Denote ∇f, $\nabla^2 f$, and $\nabla^3 f$ as the gradient, the Hessian matrix, and the third derivative tensor of f, respectively; The eigenvalues $\lambda_i(\cdot)$ for a matrix are indexed in an decreasing order with respect to their real parts, i.e., $\Re(\lambda_1(\cdot)) \geqslant \cdots \geqslant \Re(\lambda_n(\cdot))$.

Remark B1. As mentioned in [35], Assumption 1 serves as a standard requirement for analyzing the second method. The lower bound m on $\nabla^2 f_i(x)$ establishes the strong convexity of f_i, thereby ensuring that problem (2) has a unique solution. Assumption 2 can also be found in [20, 24, 31]. It will be further discussed in the following section.

Remark B2. The relationship between problem (2) and problem (1) is influenced by the penalty parameter λ. Let $x_* = (x_*^1, \ldots, x_*^n)$ denote the solution to problem (2), and let y_* denote the solution to problem (1), then it has been shown in [10] that $\|x_*^i - y_*\| = \mathcal{O}(\lambda)$ for all $i = 1, \ldots, n$. Therefore, introducing a rule to progressively decrease λ is crucial for reducing the gap between the two problems.

Appendix C Some detailed explanations

Appendix C.1 The distributed implementability of DOAOC

Define $\hat{g}_t^i \in \mathbb{R}^p$ as the ith local components of $\hat{g}_t(x_k) = (\hat{g}_t^1, \ldots, \hat{g}_t^n)$. Note that the DOAOC can be equivalently written as

$$x_{k+1} = x_k - \hat{g}_k(x_k), \tag{C1}$$

$$\hat{g}_t(x_k) = \eta\big[\nabla h(x_k) + \frac{1}{\lambda}(I_{np} - Z)x_k\big] + \big[I_{np} - \eta(\nabla^2 h(x_k) + \frac{1}{\lambda}(I_{np} - Z))\big]\hat{g}_{t-1}(x_k), \tag{C2}$$

where $t = 1, \ldots, k$ and $\hat{g}_0(x_k) = \eta\big[\nabla h(x_k) + \frac{1}{\lambda}(I_{np} - Z)x_k\big]$. Notice that $\nabla h(x_k) = \mathrm{col}\{\nabla f_1(x_k^1), \ldots, \nabla f_n(x_k^n)\}$, $\nabla^2 h(x_k) = \mathrm{diag}\{\nabla^2 f_1(x_k^1), \ldots, \nabla^2 f_n(x_k^n)\}$, and $(I_{np} - Z)x_k = \mathrm{col}\{x_k^i - \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}x_k^j\}_{i=1}^n$. By extracting the ith local component of (C1) and (C2), we obtain

$$\begin{aligned} x_{k+1}^i =& x_k^i - \hat{g}_k^i, \\ \hat{g}_t^i =& \eta\big[\nabla f_i(x_k^i) + \frac{1}{\lambda}(x_k^i - \sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}x_k^j)\big] + \big[(1-\frac{\eta}{\lambda})I_p - \eta\nabla^2 f_i(x_k^i)\big]\hat{g}_{t-1}^i + \frac{\eta}{\lambda}\sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}\hat{g}_{t-1}^j, \end{aligned}$$

where $\hat{g}_0^i = \eta\big[\nabla f_i(x_k^i) + \frac{1}{\lambda}(x_k^i - \sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}x_k^j)\big]$ and $t = 1, \ldots, k$. The update of agent i only depends on its own local information $\nabla f_i(x_k^i)$, $\nabla^2 f_i(x_k^i)$, and the variables $x_k^j, \hat{g}_{t-1}^j$ received from its neighbors. Therefore, the DOAOC algorithm can be implemented in a fully distributed manner.

Appendix C.2 Connection between DOAOC and Newton's method

The update $\hat{g}_t(x_k)$ in DOAOC can be written recursively as

$$\hat{g}_t(x_k) = \big[I_{np} - (I_{np} - \eta\nabla^2 F(x_k))^{t+1}\big]\nabla^2 F(x_k)^{-1}\nabla F(x_k),$$

which implies that if η is chosen such that $0 < \eta < \frac{2}{a}$, ensuring $\|I_{np} - \eta\nabla^2 F(x_k)\| < 1$, then $\hat{g}_t(x_k)$ converges to the Newton step as $t \to \infty$.

Appendix D Proof of Theorem 1

We now establish eigenvalue bounds for $\nabla^2 F(x)$ to facilitate the convergence proof of the proposed algorithms, as detailed below.

Proposition D1 ([24])**.** Under Assumption 1, the eigenvalues of $\nabla^2 F(x)$ are uniformly bounded as $mI_{np} \preceq \nabla^2 F(x) \preceq aI_{np}$.

The convergence of DOAOC is analyzed in the following lemma.

Lemma D1. Under Assumption 1, if set $\varepsilon = \min\{\frac{2m}{a}, \frac{1}{2}\}$, then $\{x_k\}$ generated by the DOAOC converges for any $\eta < \frac{1}{a}[1-(1-\varepsilon)^{\frac{1}{k+1}}]$.

Proof. Note that the update $\hat{g}_k(x_k)$ in (6) can be written recursively as

$$\hat{g}_k(x_k) = \big[I_{np} - (I_{np} - \eta\nabla^2 F(x_k))^{k+1}\big]\nabla^2 F(x_k)^{-1}\nabla F(x_k). \tag{D1}$$

Let $Y(x) = [I - (I - \eta\nabla^2 F(x))^{k+1}]\nabla^2 F(x)^{-1}$ and $\hat{x} = x - Y(x)\nabla F(x)$. According to Proposition D1, one has

$$\begin{aligned} F(\hat{x}) \leqslant& F(x) - \nabla F(x)^\top Y(x)\nabla F(x) + \frac{a}{2}(Y(x)\nabla F(x))^\top Y(x)\nabla F(x) \\ =& F(x) - \nabla F(x)^\top [Y(x) - \frac{a}{2}Y(x)^\top Y(x)]\nabla F(x). \end{aligned} \tag{D2}$$

Since $\nabla^2 F(x) \succ \mathbf{0}_{np}$, there exists a nonsingular matrix P such that $\nabla^2 F(x) = P^{-1}DP$, where $D = \mathrm{diag}\{d_1, d_2, \cdots, d_{np}\}$ is a positive definite diagonal matrix. Then

$$\begin{aligned} Y(x) =& [I_{np} - (I_{np} - \eta\nabla^2 F(x))^{k+1}]\nabla^2 F(x)^{-1} \\ =& P^{-1}[D^{-1} - (I_{np} - \eta D)^{k+1}D^{-1}]P. \end{aligned} \tag{D3}$$

Without loss of the generality, assume $0 < d_1 \leqslant d_2 \leqslant \cdots \leqslant d_{np}$. In this case,

$$\begin{aligned}
&2m(Y(x) - \frac{a}{2}Y(x)^\top Y(x)) \\
=&P^{-1}2m[(D^{-1} - (I_{np} - \eta D)^{k+1}D^{-1}) - \frac{a}{2}(D^{-1} - (I_{np} - \eta D)^{k+1}D^{-1})^2]P \\
=&P^{-1}2m \cdot \mathrm{diag}\{\frac{1}{d_i}[1 - (1 - \eta d_i)^{k+1}] - \frac{a}{2d_i^2}[1 - (1 - \eta d_i)^{k+1}]^2\}_{i=1}^{np}P.
\end{aligned} \tag{D4}$$

When $0 < 1 - \eta d_i < 1$, there hold: 1) $0 < 1 - (1 - \eta d_i)^{k+1} < 1$ for any $\eta \in (0, +\infty)$; 2) $1 - (1 - \eta d_i)^{k+1}$ is a increasing function over η in $(0, +\infty)$; 3) $\lim_{\eta \to 0}[1 - (1 - \eta d_i)^{k+1}] = 0$. These facts means for any $0 < \varepsilon < 1$, there always exists a $\bar{\eta} > 0$ such that when $\eta \leqslant \bar{\eta}$, there always holds $0 < 1 - (1 - \eta d_i)^{k+1} < \varepsilon$. Let $\varepsilon = \min\{\frac{2d_i}{a}, \frac{d_i}{2m}, i = 1, \cdots, np\} = \min\{\frac{2m}{a}, \frac{1}{2}\} \leqslant \frac{1}{2}$. Then $0 < \frac{1}{\varepsilon}[1 - (1 - \eta d_i)^{k+1}] < 1$. Accordingly, we can derive $0 < \frac{2m}{d_i}[1 - (1 - \eta - d_i)^{k+1}] < 1$ and $0 < 1 - \frac{a}{2d_i}[1 - (1 - \eta d_i)^{k+1}] < 1$. Denote

$$\mu_i(\eta) = 2m\{\frac{1}{d_i}[1 - (1 - \eta d_i)^{k+1}] - \frac{a}{2d_i^2}[1 - (1 - \eta d_i)^{k+1}]^2\}. \tag{D5}$$

It is immediate to get $0 < \mu_i(\eta) = \{\frac{2m}{d_i}[1 - (1 - \eta d_i)^{k+1}]\}\{1 - \frac{a}{2d_i}[1 - (1 - \eta d_i)^{k+1}]\} < 1$. The above analysis shows that any $0 < \eta \leqslant \bar{\eta}$ is a safe choice for $0 < \mu_i(\eta) < 1$. Also, we can find an upper bound for $\bar{\eta}$. Solve η from the inequality $1 - (1 - \eta d_i)^{k+1} < \varepsilon$. We can acquire $\eta < \frac{1}{d_i}(1 - (1 - \varepsilon)^{\frac{1}{k+1}})$. Consequently, $\eta < \min\{\frac{1}{a}[1 - (1 - \varepsilon)^{\frac{1}{k+1}}], \frac{1}{a}\}$ can guarantee $0 < \mu_i(\eta) < 1$ and $0 < 1 - \mu_i(\eta) < 1$ for $i = 1, \cdots, np$. From (D2), $F(\hat{x}) - F(x_*) \leqslant (1 - \min\{\mu_i(\eta), i = 1, \cdots, np\})(F(x) - F(x_*))$. Therefore, DOAOC is convergent for $0 < \eta < \min\ \frac{1}{a}[1 - (1 - \varepsilon)^{\frac{1}{k+1}}], \frac{1}{a}\} = \frac{1}{a}[1 - (1 - \varepsilon)^{\frac{1}{k+1}}]$.

With the preparation in Lemma D1 and Proposition D1, we turn to prove Theorem 1.

Proof. By direct derivation, we obtain

$$\begin{aligned}
\hat{g}_k(x_*) =&[I_{np} - (I_{np} - \eta\nabla^2 F(x_*))^{k+1}]\nabla^2 F(x_*)^{-1}\nabla F(x_*) \\
=&\mathbf{0}_{np}, \\
\nabla\hat{g}_k(x_*) =&I_{np} - (I_{np} - \eta\nabla^2 F(x_*))^{k+1}.
\end{aligned}$$

It follows from (6) that

$$\begin{aligned}
&||x_{k+1} - x_*|| \\
=&||x_k - x_* - \hat{g}_k(x_k)|| \\
=&||x_k - x_* - [\hat{g}_k(x_*) + \nabla\hat{g}_k(x_*)(x_k - x_*) + r_k]|| \\
\leqslant&||x_k - x_* - \nabla\hat{g}_k(x_*)(x_k - x_*)|| + ||r_k|| \\
=&||(I_{np} - \eta\nabla^2 F(x_*))^{k+1}(x_k - x_*)|| + ||r_k|| \\
\leqslant&||I_{np} - \eta\nabla^2 F(x_*)||^{k+1}||x_k - x_*|| + ||r_k||.
\end{aligned}$$

Since $||I_{np} - \eta\nabla^2 F(x_*)|| < 1$ when $\eta < \frac{2}{a}$, one has $\lim_{k\to\infty} \frac{||x_{k+1} - x_*||}{||x_k - x_*||} = 0$, i.e., the DOAOC is superlinearly convergent.

Appendix E Proof of Theorem 2

To lay the theoretical foundation for the convergence analysis of DOAOC-K, we first present the following lemma.

Lemma E1. For the function $F(x)$ in (2), if Assumptions 1 and 2 are satisfied, then for any $x, \hat{x} \in \mathbb{R}^{np}$, it holds that

$$F(\hat{x}) \leqslant F(x) + \nabla F(x)^\top(\hat{x} - x) + \frac{1}{2}(\hat{x} - x)^\top\nabla^2 F(x)(\hat{x} - x) + \frac{L}{6}||\hat{x} - x||^3. \tag{E1}$$

Proof. Let $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^n)$ and $x = (x^1, \ldots, x^n)$. Under Assumption 2, we have

$$\begin{aligned}
||\nabla^2 F(\hat{x}) - \nabla^2 F(x)|| =& \max_{\mathrm{i}=1,\cdots\mathrm{n}} ||\nabla^2 f_i(\hat{x}^i) - \nabla^2 f_i(x^i)|| \\
\leqslant& L \max_{\mathrm{i}=1,\cdots\mathrm{n}} ||\hat{x}^i - x^i|| \leqslant L||\hat{x} - x||,
\end{aligned} \tag{E2}$$

which implies that the third derivative tensor of $F(x)$ is bounded in norm by L. Then, the second-order Taylor expansion of $F(\hat{x})$ at x is given by

$$F(\hat{x}) = F(x) + \nabla F(x)^\top(\hat{x} - x) + \frac{1}{2}(\hat{x} - x)^\top\nabla^2 F(x)(\hat{x} - x) + E_2(x, \hat{x}), \tag{E3}$$

where $E_2(x, \hat{x})$ is the remainder term. The subsequent proof can be derived directly using (E2) and the Taylor expansion error as discussed in Theorem 7.6 of [36], i.e.,

$$\begin{aligned}
&|E_2(x, \hat{x})| \\
\leqslant& \int_0^1 \frac{(1 - s)^2}{2} \cdot \|\nabla^3 F(x + s(\hat{x} - x))\| \cdot \|\hat{x} - x\|^3 ds \\
\leqslant& L\int_0^1 \frac{(1 - s)^2}{2} ds \cdot ||\hat{x} - x||^3 \\
=& \frac{L}{6}||\hat{x} - x||^3.
\end{aligned} \tag{E4}$$

Substituting (E4) into (E3) completes the proof.

In the subsequent part, we prove that by using DOAOC-K, the function value $F(x_k)$ gradually converges to $F(x_*)$ as the iterations progress. To illustrate this, we establish an upper bound for $F(x_{k+1}) - F(x_*)$ using $F(x_k) - F(x_*)$, which serves as a basis for establishing the convergence of DOAOC-K.

Lemma E2. Under Assumptions 1 and 2, if select the step size as

$$\eta < \min\left\{1, \frac{1}{a}, \frac{2m}{a^2K^2}\right\}, \tag{E5}$$

then $\{x_k\}$ generated by the DOAOC-K algorithm satisfies

$$F(x_{k+1}) - F(x_*) \leqslant \left[1 - \frac{2m^2\eta - ma^2\eta^2K^2}{a}\right](F(x_k) - F(x_*)) + \frac{a^3(2a)^{\frac{3}{2}}\eta^3K^3L}{6m^3}(F(x_k) - F(x_*))^{\frac{3}{2}}. \tag{E6}$$

Proof. The DOAOC-K can be recursively written as $x_{k+1} = x_k - H\nabla^2F(x_k)^{-1}\nabla F(x_k)$, where $H = I_{np} - (I_{np} - \eta\nabla^2F(x_k))^K$. First, we shall show that $H\nabla^2F(x)^{-1}$ is positive definite. Assuming that $\eta < \frac{1}{a}$ and using the Bernoulli inequality $(1-\mu)^k \geqslant 1 - k\mu$ for all $\mu \in (0,1)$ [37], it can be easily shown that

$$\begin{aligned}
&\lambda_1(H) \leqslant 1 - (1 - \eta K\lambda_1(\nabla^2F(x_k))) \leqslant a\eta K,\\
&\lambda_n(H) \geqslant 1 - (1 - \eta\lambda_n(\nabla^2F(x_k))) \geqslant m\eta > 0.
\end{aligned}$$

It can be verified from

$$\begin{aligned}
&H\nabla^2F(x)^{-1}\\
=&\nabla^2F(x)^{-1} - (I_{np} - \eta\nabla^2F(x))(I_{np} - \eta\nabla^2F(x))\cdots(I_{np} - \eta\nabla^2F(x))\nabla^2F(x)^{-1}\\
=&\nabla^2F(x)^{-1} - (I_{np} - \eta\nabla^2F(x))(I_{np} - \eta\nabla^2F(x))\cdots\nabla^2F(x)^{-1}(I_{np} - \eta\nabla^2F(x))\\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots\\
=&\nabla^2F(x)^{-1} - \nabla^2F(x)^{-1}(I_{np} - \eta\nabla^2F(x))(I_{np} - \eta\nabla^2F(x))\cdots(I_{np} - \eta\nabla^2F(x))\\
=&\nabla^2F(x)^{-\top}H^\top
\end{aligned}$$

and $\lambda_n(H\nabla^2F(x)^{-1}) > \lambda_n(H)\lambda_n(\nabla^2F(x)^{-1}) > 0$ that $H\nabla^2F(x)^{-1} \succ 0$. We can also obtain that $\nabla^2F(x)H \succ 0$ by using a similar method. Further, by setting $\hat{x} \triangleq x_{k+1}$ and $x \triangleq x_k$ in Lemma E1, it follows that

$$\begin{aligned}
&F(x_{k+1})\\
\leqslant&F(x_k) + \nabla F(x_k)^\top(x_{k+1} - x_k) + \frac{1}{2}(x_{k+1} - x_k)^\top\nabla^2F(x_k)(x_{k+1} - x_k) + \frac{L}{6}||x_{k+1} - x_k||^3\\
=&F(x_k) - \nabla F(x_k)^\top H\nabla^2F(x_k)^{-1}\nabla F(x_k)\\
&+ \frac{1}{2}\nabla F(x_k)^\top\nabla^2F(x_k)^{-T}h^\top\nabla^2F(x_k)H\nabla^2F(x_k)^{-1}\nabla F(x_k) + \frac{L}{6}||H\nabla^2F(x_k)^{-1}\nabla F(x_k)||^3\\
\leqslant&F(x_k) - m\eta\nabla F(x_k)^\top\nabla^2F(x_k)^{-1}\nabla F(x_k) + \frac{a^2\eta^2K^2}{2}\nabla F(x_k)^\top\nabla^2F(x_k)^{-1}\nabla F(x_k)\\
&+ \frac{a^3\eta^3K^3L}{6}||\nabla^2F(x_k)^{-1}\nabla F(x_k)||^3\\
\leqslant&F(x_k) - \frac{2m\eta - a^2\eta^2K^2}{2a}\nabla F(x_k)^\top\nabla F(x_k) + \frac{a^3\eta^3K^3L}{6m^3}||\nabla F(x_k)||^3.
\end{aligned} \tag{E7}$$

According to Proposition D1, for any $x, \hat{x} \in \mathbb{R}^{np}$, there holds

$$F(\hat{x}) \geqslant F(x) + \nabla F(x)^\top(\hat{x} - x) + \frac{m}{2}(\hat{x} - x)^\top(\hat{x} - x), \tag{E8}$$

$$F(\hat{x}) \leqslant F(x) + \nabla F(x)^\top(\hat{x} - x) + \frac{a}{2}(\hat{x} - x)^\top(\hat{x} - x). \tag{E9}$$

It can be proven from (E8) and (E9) that

$$F(x_*) \geqslant F(x_k) - \frac{1}{2m}\nabla F(x_k)^\top\nabla F(x_k), \tag{E10}$$

$$F(x_*) \leqslant F(x_k) - \frac{1}{2a}\nabla F(x_k)^\top\nabla F(x_k). \tag{E11}$$

Setting $\eta < \frac{2m}{a^2K^2}$ and substituting (E10) and (E11) into (E7), one has

$$F(x_{k+1}) - F(x_*) \leqslant \left[1 - \frac{2m^2\eta - ma^2\eta^2K^2}{a}\right](F(x_k) - F(x_*)) + \frac{a^3(2a)^{\frac{3}{2}}\eta^3K^3L}{6m^3}(F(x_k) - F(x_*))^{\frac{3}{2}}.$$

Based on the results developed in Lemma E2, we now prove Theorem 2.

Proof. From Lemma E2, we can rewrite (E6) as

$$F(x_{k+1}) - F(x_*) \leqslant (1-\beta_k)(F(x_k) - F(x_*)), \tag{E12}$$

where

$$\beta_k = \frac{2m^2\eta - ma^2\eta^2K^2}{a} - \frac{a^3(2a)^{\frac{3}{2}}\eta^3K^3L}{6m^3}(F(x_k) - F(x_*))^{\frac{1}{2}}. \tag{E13}$$

It remains to show that β_k satisfies $0 < \beta_k < 1$ for all indices k. We first prove $\beta_k < 1$. It is easy to find

$$\begin{aligned}\beta_k \leqslant & \frac{2m^2\eta - ma^2\eta^2K^2}{a} \\ \leqslant & \frac{2ma\eta K - ma^2\eta^2K^2}{a} \\ \leqslant & \frac{m}{a} < 1,\end{aligned}$$

where the second inequality holds since $K \geqslant 1$ and $m < a$. We now turn to prove that $\beta_k > 0$. Let $\eta < \frac{m}{a^2K^2}$, then we have

$$\begin{aligned}\beta_0 = & \frac{2m^2\eta - ma^2\eta^2K^2}{a} - \frac{a^3(2a)^{\frac{3}{2}}\eta^3K^3L}{6m^3}(F(x_0) - F(x_*))^{\frac{1}{2}} \\ = & \frac{m^2\eta}{a} - \frac{a^3(2a)^{\frac{3}{2}}\eta^3K^3L}{6m^3}(F(x_0) - F(x_*))^{\frac{1}{2}} + \frac{m^2\eta - ma^2\eta^2K^2}{a} \\ \geqslant & \frac{m^2\eta}{a} - \frac{a^3(2a)^{\frac{3}{2}}\eta^3K^3L}{6m^3}(F(x_0) - F(x_*))^{\frac{1}{2}},\end{aligned} \tag{E14}$$

where the last inequality comes from $m^2\eta - ma^2\eta^2K^2 > 0$. According to (E14), we can conclude that $\beta_0 > 0$ by setting $\eta < \left[\frac{6m^5}{a^4(2a)^{\frac{2}{3}}K^3L(F(x_0)-F(x_*))^{\frac{1}{2}}}\right]^{\frac{1}{2}}$. It then follows immediately from (E12) and (E13) that $F(x_1) - F(x_*) < F(x_0) - F(x_*)$ and $\beta_0 < \beta_1$. Using a similar method, we can also establish that $0 < \beta_k < \beta_{k+1} < 1$ for all indices k. Setting $\bar{\epsilon} = \beta_0$, the proof is completed.

Appendix F Numerical simulation

Appendix E offers the pseudocodes of the developed DOAOC and DOAOC-K algorithms and compares them with other second-order methods based on the penalty function interpretation, such as Network Newton (NN-K) from [24] and distributed quasi-Newton (DQN-l) from [25], for minimizing distributed quadratic objective functions. We also conduct a comparison between the proposed methods and traditional first-order approaches, including DGD and its accelerated form of DGD (ACC-DGD) described in [12].

Appendix F.1 The distributed implementation of the proposed algorithms

Now we summarize the DOAOC from the perspective of each agent i, refer to Algorithm F1 for details. We begin by setting the parameters $\eta, \lambda > 0$, and initializing $x_0^i \in \mathbb{R}^p$ of agent i in Step 1. In Step 3, agent i sends x_k^i to its neighbors $j \in \mathcal{N}_i$ and receives x_k^j from them, ensuring that each agent has the local information associated with its neighbors for subsequent updates. In Step 4, agent i initializes the loop by computing the initial $\hat{g}_0^i$ based on the local information. Steps 6 and 7 require receiving $\hat{g}_t^j$ from neighbors to obtain the update size for Step 9. It is worth mentioning that the communication of DOAOC is restricted to neighboring nodes, as detailed in Steps 3 and 6.

Algorithm F1 DOAOC distributed implementation.

1: Initialization: Each agent i requires $\eta, \lambda > 0$ and sets $x_0^i \in \mathbb{R}^p$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Each agent i sends x_k^i to all neighbors $j \in \mathcal{N}_i$ and receives x_k^j from them.
4: Each agent i computes $\hat{g}_0^i = \eta\big[\nabla f_i(x_k^i) + \frac{1}{\lambda}(x_k^i - \sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}x_k^j)\big]$.
5: **for** $t = 0, \ldots, k-1$ **do**
6: Each agent i receives $\hat{g}_t^j$ with neighbors $j \in \mathcal{N}_i$.
7: Each agent i updates $\hat{g}_{t+1}^i = \eta\big[\nabla f_i(x_k^i) + \frac{1}{\lambda}(x_k^i - \sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}x_k^j)\big] + (1-\frac{\eta}{\lambda})\hat{g}_t^i - \eta\nabla^2 f_i(x_k^i)\hat{g}_t^i + \frac{\eta}{\lambda}\sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}\hat{g}_t^j$.
8: **end for**
9: Each agent i updates $x_{k+1}^i = x_k^i - \hat{g}_k^i$.
10: **end for**

Given the similarities between DOAOC-K and DOAOC, we will not provide further details on DOAOC-K. For any integer parameter $K > 1$, the distributed version of DOAOC-K is presented in Algorithm F2.

Algorithm F2 DOAOC-K distributed implementation.

Initialization: Each agent i requires $\eta, \lambda > 0$, positive integer $K > 1$ and sets $x_0^i \in \mathbb{R}^p$.
for $k = 0, 1, \ldots$ **do**
  Each agent i sends x_k^i to all neighbors $j \in \mathcal{N}_i$ and receives x_k^j from them.
  Each agent i computes $\hat{g}_0^i = \eta\big[\nabla f_i(x_k^i) + \frac{1}{\lambda}(x_k^i - \sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}x_k^j)\big]$.
  for $t = 0, \ldots, K-2$ **do**
    Each agent i receives $\hat{g}_t^j$ with neighbors $j \in \mathcal{N}_i$.
    Each agent i updates $\hat{g}_{t+1}^i = \eta\big[\nabla f_i(x_k^i) + \frac{1}{\lambda}(x_k^i - \sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}x_k^j)\big] + (1-\frac{\eta}{\lambda})\hat{g}_t^i - \eta\nabla^2 f_i(x_k^i)\hat{g}_t^i + \frac{\eta}{\lambda}\sum_{j\in\mathcal{N}_i\cup\ i\}} w_{ij}\hat{g}_t^j$.
  end for
  Each agent i updates $x_{k+1}^i = x_k^i - \hat{g}_{K-1}^i$.
end for

Appendix F.2 Comparison with existing methods

Considering that each agent is endowed with a distinct local objective function $f_i(y) = \frac{1}{2}y^\top A_i y + b_i^\top y$, where $b_i \in \mathbb{R}^p$ is drawn from the normal distribution. To generate $A_i \in \mathbb{S}_{++}^p$, we start with a matrix $\hat{A}_i$, whose entries are independently sampled from the standard normal distribution. The matrix A_i is then constructed as $A_i = \hat{A}_i\hat{A}_i^\top$. The agents collectively aim to find the solution y_* of the objective function $f(y) = \sum_{i=1}^n \frac{1}{2}y^\top A_i y + b_i^\top y$.

The connected network is constructed by randomly generating $\frac{\tau n(n-1)}{2}$ undirected edges among n nodes, where $\tau \in (0, 1]$ denotes the connectivity ratio. This parameter τ determines the proportion of possible edges that are actually present in the network, and is chosen so as to ensure that the resulting graph remains fully connected, enabling message exchange between any two nodes through one or more paths. We start by creating a cycle that connects all nodes, then randomly adding the remaining edges between pairs of nodes that do not have a direct connection. Using the Sinkhorn-Knopp algorithm in conjunction with the network topology, we generate a doubly stochastic matrix W. The relative error at the kth iteration takes the form

$$e \triangleq \frac{1}{n}\sum_{i=1}^n \frac{||x_k^i - y_*||}{||y_*||},$$

where y_* is pre-computed through centralized Newton's method. We consider $n = 20$, $p = 5$, $\tau = 0.3$, initializing the variables for all agents $i = 1, \ldots, n$ as $x_0^i = \mathbf{0}_p$, where the distance from y_* is larger than 50. The penalty coefficient is set to $\lambda = 10^{-3}$.

We first randomly generate an objective function and construct a network topology, and then we compare the performance of these algorithms within this framework. We set $K = 3$ for DOAOC-K, $l = 2$ for DQN-l, and $K = 2$ for NN-K. Notably, DQN-2 and NN-2 are the best-performing variants among DQN-l ($l = 0, 1, 2$) and NN-K ($K = 0, 1, 2$), respectively. The reason for selecting DOAOC-3 is that its communication per iteration is the same as that of DQN-2 and NN-2. For the DQN-2 method, the parameters (δ, ϵ) are selected based on the recommendations from [25], i.e., $\delta = 0$ and $\epsilon = 1$. For the NN-2 method, we set the step size $\epsilon = 2$, which is the hand-optimized value. For ACC-DGD and DGD, it is supposed that the step size and momentum coefficients remain unchanged throughout the iterations. This setting allows a fair assessment of ACC-DGD, DGD, and DOAOC, with the aim of examining their effectiveness in handling the penalty problem. We set $\alpha = 0.001$ and $\beta = 0.3$ for ACC-DGD where $\beta = 0.3$ is found to be the optimal choice among $\{0.1, 0.2, \ldots, 0.9\}$, and $\alpha = 0.001$ for DGD. The step size for both the DOAOC and DOAOC-K algorithms is set to $\eta = 0.0013$.

Figure F1 depicts the numerical results on the relation between relative error and iterations. The results illustrate that DOAOC reaches proximity to the optimal solution within 50 iterations, followed by DOAOC-3, NN-2, and DQN-2. For first-order methods, we can see that ACC-DGD is faster than DGD, but both are slower than second-order methods. Notably, DOAOC requires considerably fewer iterations compared with first-order methods, and both DOAOC and DOAOC-K algorithms exhibit more favorable convergence behavior.

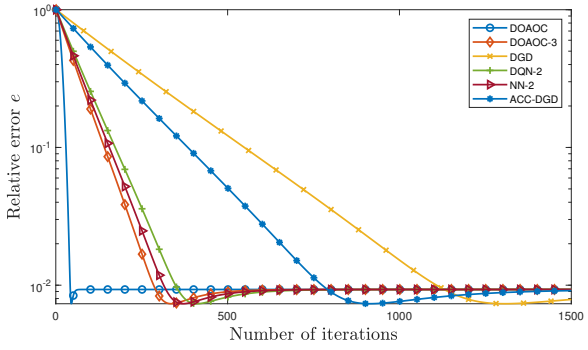

Figure F1 Relative error e of DOAOC, DOAOC-3, NN-2, DQN-2, ACC-DGD, and DGD versus the number of iterations.

To more clearly demonstrate the superiority of the proposed algorithms over the other algorithms, we have summarized their differences in terms of convergence ratio and numerical performance. Following the explanation of convergence ratio r in [38], we define

the convergence ratio r is defined as

$$r = \frac{N_{\text{alg}}}{N_{\text{DOAOC}}},$$

where N_{alg} denotes the number of iterations required by a given algorithm to reach a prescribed accuracy, and N_{DOAOC} denotes the number of iterations required by DOAOC to reach the same accuracy. As shown in Table F1 in the response, DOAOC reaches the target accuracy in the fewest iterations. In contrast, the convergence ratios of other second-order methods are all greater than 1, while the convergence ratio of DOAOC-K is lower than that of both other second-order and first-order methods. First-order methods exhibit the highest convergence ratios, reflecting the limitations in their use of available information.

Table F1 Comparison of convergence between the proposed methods and baselines when the relative error reaches 10^{-2}.

Algorithm	Type	Convergence rate	Iterations	Convergence ratio r
DOAOC	2nd-order	Superlinear	Lowest (42)	1.00
DOAOC-3	2nd-order	Linear	Low (285)	6.79
NN-2	2nd-order	Linear	Medium (314)	7.48
DQN-2	2nd-order	Linear	Medium (349)	8.31
DGD	1st-order	Linear	High (1109)	26.4
ACC-DGD	1st-order	Linear	High (777)	18.5

We conduct 1000 random trials with the same parameters as mentioned above and analyze the statistical distribution of communication rounds required to achieve an accuracy of $e = 10^{-2}$, as illustrated in Figure F2. From the results, one can see that the average number of communication rounds for DOAOC and DOAOC-3 is 869 and 685, while the average numbers for NN-2, DQN-2, ACC-DGD, and DGD are 970, 1096, 1186, and 912, respectively. It can be observed that DOAOC requires more communication compared to DOAOC-3, but both DOAOC and DOAOC-3 have fewer communication rounds than the other algorithms. Compared to DGD, the communication rounds are reduced by approximately 25% and 40% for DOAOC and DOAOC-K, respectively. In particular, the number of communication rounds can be accepted if DOAOC converges rapidly. Compared with the DOAOC, the DOAOC-3 has a slower convergence rate but requires fewer communications, verifying the effectiveness of our proposed DOAOC-K in balancing iterations and communication rounds.

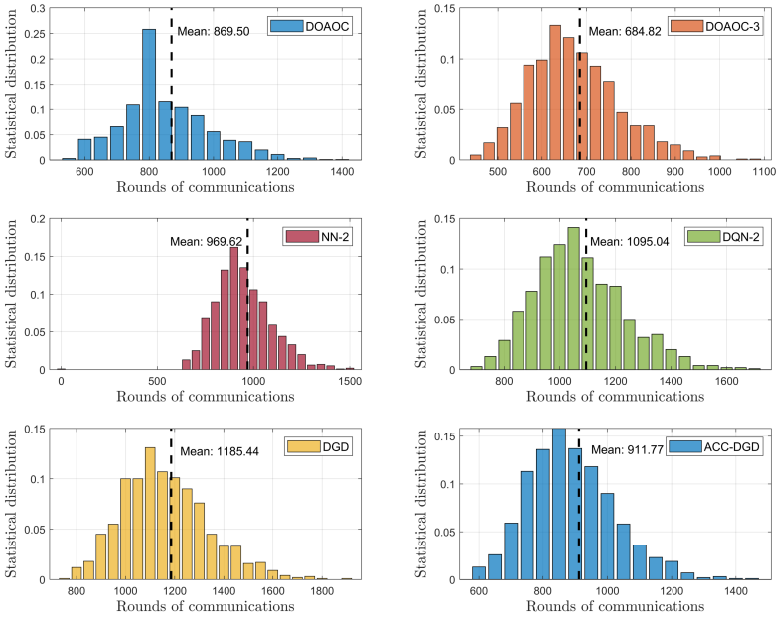

Figure F2 Histograms of rounds of communications in 1000 trials.

References

1 Khan U A, Kar S, Moura J M. DILAND: An algorithm for distributed sensor localization with noisy distance measurements. IEEE Trans Signal Process, 2009, 58(3): 1940–1947

2 Rabbat M, Nowak R. Distributed optimization in sensor networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA, 2004. 20–27

3 Schizas I D, Ribeiro A, Giannakis G B. Consensus in ad hoc WSNs with noisy links–Part I: Distributed estimation of deterministic signals. IEEE Trans Signal Process, 2007, 56(1): 350–364

4 Mota J F, Xavier J M, Aguiar P M, et al. Distributed optimization with local domains: applications in MPC and network flows. IEEE Trans Autom Control, 2014, 60(7): 2004–2009
5 Cao Y C, Yu W, Ren W W, et al. An overview of recent progress in the study of distributed multi-agent coordination. IEEE Trans Ind Informat, 2012, 9(1): 427–438
6 Lopes C G, Sayed A H. Diffusion least-mean squares over adaptive networks: formulation and performance analysis. IEEE Trans Signal Process, 2008, 56(7): 3122–3136
7 Koppel A, Paternain S, Richard C, et al. Decentralized online learning with kernels. IEEE Trans Signal Process, 2018, 66(12): 3240–3255
8 Lee D, He N, Kamalaruban P, et al. Optimization for reinforcement learning: from a single agent to cooperative agents. IEEE Signal Process Mag, 2020, 37(3): 123–135
9 Bedi A S, Koppel A, Rajawat K. Asynchronous online learning in multi-agent systems with proximity constraints. IEEE Trans Signal Inf Process Netw, 2019, 5(3): 479–494
10 Yuan K, Ling Q, Yin W T. On the convergence of decentralized gradient descent. SIAM J Optim, 2016, 26(3): 1835–1854
11 Nedic A, Ozdaglar A. Distributed subgradient methods for multi-agent optimization. IEEE Trans Autom Control, 2009, 54(1): 48–61
12 Jakovetić D, Xavier J, Moura J M. Fast distributed gradient methods. IEEE Trans Autom Control, 2014, 59(5): 1131–1146
13 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn, 2011, 3(1): 1–122
14 Mokhtari A, Shi W, Ling Q, et al. DQM: Decentralized quadratically approximated alternating direction method of multipliers. IEEE Trans Signal Process, 2016, 64(19): 5158–5173
15 Chang T-H, Hong M Y, Wang X F. Multi-agent distributed optimization via inexact consensus ADMM. IEEE Trans Signal Process, 2014, 63(2): 482–497
16 Duchi J C, Agarwal A, Wainwright M J. Dual averaging for distributed optimization: convergence analysis and network scaling. IEEE Trans Autom Control, 2011, 57(3): 592–606
17 Nesterov Y. Lectures on Convex Optimization. Berlin: Springer, 2018
18 Zhang H S, Wang H X, Xu Y M, et al. Optimization methods rooted in optimal control. Sci China Inf Sci, 2024, 67(12): 1–9
19 Lv C Z, Yin X M, Li H D, et al. Optimal Control of Nonlinear Systems Using Accelerated Gradient and Hessian Computation. IEEE Trans Ind Electron, 2025, 1–9
20 Zhang J Q, You K Y, Başar T. Distributed adaptive Newton methods with global superlinear convergence. Automatica, 2022, 138: 110156
21 Shamir O, Srebro N, Zhang T. Communication-efficient distributed optimization using an approximate Newton-type method. In: Proceedings of the International Conference on Machine Learning, Beijing, China, 2014. 1000–1008
22 Wang S S, Roosta F, Xu P, et al. GIANT: Globally improved approximate Newton method for distributed optimization. Adv Neural Inf Process Syst, 2018, 31
23 Soori S, Mishchenko K, Mokhtari A, et al. DAve-QN: A distributed averaged quasi-Newton method with local superlinear convergence rate. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 2020. 1965–1976
24 Mokhtari A, Ling Q, Ribeiro A. Network Newton distributed optimization methods. IEEE Trans Signal Process, 2016, 65(1): 146–161
25 Bajovic D, Jakovetic D, Krejic N, et al. Newton-like method with diagonal correction for distributed optimization. SIAM J Optim, 2017, 27(2): 1171–1203
26 Qu Z H, Li X X, Li L, Hong Y G. Linearly convergent second-order distributed optimization algorithms. IEEE Trans Autom Control, 2024, 69(8): 5431–5438
27 Mokhtari A, Shi W, Ling Q, et al. A decentralized second-order method with exact linear convergence rate for consensus optimization. IEEE Trans Signal Inf Process Netw, 2016, 2(4): 507–522
28 Eisen M, Mokhtari A, Ribeiro A. A primal-dual Quasi-Newton method for exact consensus optimization. IEEE Trans Signal Process, 2019, 67(23): 5983–5997
29 Ghalkha A, Issaid C B, Elgabli A, Bennis M. DIN: A decentralized inexact Newton algorithm for consensus optimization. IEEE Trans Mach Learn Commun Netw, 2024, 2: 663–674
30 Varagnolo D, Zanella F, Cenedese A, et al. Newton-Raphson consensus for distributed convex optimization. IEEE Trans Autom Control, 2015, 61(4): 994–1009
31 Zhang J J, Ling Q, So A M-C. A Newton tracking algorithm with exact linear convergence for decentralized consensus optimization. IEEE Trans Signal Inf Process Netw, 2021, 7: 346–358
32 Ye H S, He S Y, Chang X Y. DINE: Decentralized Inexact Newton With Exact Linear Convergence Rate. IEEE Trans Signal Process, 2023, 72: 143–156
33 Jakovetić D, Krejić N, Malaspina G. Distributed inexact Newton method with adaptive step sizes. Comput Optim Appl, 2025, 91(2): 683–715
34 Niu D B, Hong Y G, Song E B. A dual inexact nonsmooth Newton method for distributed optimization. IEEE Trans Signal Process, 2024, 73, 188–203
35 Boyd S, Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press, 2004
36 Apostol T M. Calculus, volume 61. Hoboken: Wiley, 1991
37 Mitrinovic D S, Pecaric J, Fink A M. Classical and new inequalities in analysis, volume 61. Amsterdam: Springer, 2013
38 Dolan E D, Moré J J. Benchmarking optimization software with performance profiles. Math Program, 2002, 91(2): 201–213