SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2024, Vol. 67, Iss. 11, 212201:1–212201:16
https://doi.org/10.1007/s11432-023-4086-5

Improved dynamic regret of distributed online multiple Frank-Wolfe convex optimization

Wentao ZHANG[1], Yang SHI[2], Baoyong ZHANG[1*] & Deming YUAN[1]

[1]*School of Automation, Nanjing University of Science and Technology, Nanjing* 210094*, China;*
[2]*Department of Mechanical Engineering, University of Victoria, Victoria* V8W 2Y2*, Canada*

Received 15 November 2023/Revised 3 March 2024/Accepted 5 May 2024/Published online 9 October 2024

Abstract In this paper, we explore a distributed online convex optimization problem over a time-varying multi-agent network. The network aims to minimize a global loss function through local computation and communication with neighboring agents. To effectively handle the optimization problem which involves high-dimensional and structural constraint sets, we develop a distributed online multiple Frank-Wolfe algorithm that circumvents the expensive computational cost associated with projection operations. The dynamic regret bounds are established as $\mathcal{O}(T^{1-\gamma} + H_T)$ with the linear oracle number $\mathcal{O}(T^{1+\gamma})$, which depends on the horizon (total iteration number) T, the function variation H_T, and the tuning parameter $0 < \gamma < 1$. In particular, when the prior knowledge of H_T and T is available, the bound can be enhanced to $\mathcal{O}(1 + H_T)$. Moreover, we explore the significant advantages provided by the multiple iteration technique and reveal a trade-off between dynamic regret bound, computational cost, and communication cost. Finally, the performance of our algorithm is validated and compared through the distributed online ridge regression problems with two constraint sets.

Keywords distributed online convex optimization, multiple iterations, Frank-Wolfe algorithm, dynamic regret, gradient tracking method

1 Introduction

Online convex optimization (OCO) [1,2] has emerged as a powerful paradigm for learning and has recently garnered significant attention in some complex scenarios of optimization and machine learning [3–13]. In this framework, the decision maker first makes an action at each round, suffers a loss from the environment or adversary, and then receives information about the loss function to update the next action. The main goal is to minimize the accumulated loss over time.

Over the past two decades, research in distributed optimization has increased rapidly [14–25] owing to its prominent advantages such as low computational burden, robustness compared to centralized structures, and wide-ranging applications including sensor networks, signal processing, smart grids, and machine learning. Early studies [7, 8] incorporating the OCO framework presented the convergence analysis for distributed online optimization and established an $\mathcal{O}(\sqrt{T})$ regret for convex loss functions. Following this, various distributed algorithms suitable for the OCO framework have been developed, such as distributed online mirror descent [4, 10] and distributed online push-sum [6]. This paper addresses the following distributed optimization problem under the OCO framework:

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T} \sum_{t=1}^{T} F_t(\boldsymbol{x}_t), \text{ s.t. } \boldsymbol{x}_t \in \boldsymbol{X}, \tag{1}$$

where T is the time horizon, $\boldsymbol{X} \in \mathbb{R}^d$ is a convex compact set, the function $F_t(\boldsymbol{x}_t) = \sum_{i=1}^{n} f_{i,t}(\boldsymbol{x}_t)$, and $f_{i,t}$ is convex in $\boldsymbol{X}$. Static regret and dynamic regret are commonly used as performance metrics to measure online optimization algorithms. Static regret $\mathrm{Regret}_s^j(T)$ [26] shown in (2) represents the

* Corresponding author (email: baoyongzhang@njust.edu.cn)

difference between the cumulative loss incurred from the decision sequence $\{\boldsymbol{x}_{j,t}\}$ of the agent j over time T and the total loss at the optimal benchmark $\boldsymbol{x}^*$:

$$\mathrm{Regret}_s^j(T) = \sum_{t=1}^{T} F_t(\boldsymbol{x}_{j,t}) - \sum_{t=1}^{T} F_t(\boldsymbol{x}^*), \tag{2}$$

where $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\boldsymbol{X}} \sum_{t=1}^{T} F_t(\boldsymbol{x})$. By contrast, as a more stringent metric, dynamic regret [26, 27] defined as

$$\mathrm{Regret}_d^j(T) = \sum_{t=1}^{T} F_t(\boldsymbol{x}_{j,t}) - \sum_{t=1}^{T} F_t(\boldsymbol{x}_t^*), \tag{3}$$

where $\boldsymbol{x}_t^* \in \arg\min_{\boldsymbol{x}\in\boldsymbol{X}} F_t(\boldsymbol{x})$, accurately reflects decision quality in many applications, such as the objective tracking with multiple robots [28], owing to its varying benchmark sequence $\{\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_T^*\}$.

The bound of dynamic regret typically relies on certain regularities of the optimization problem and does not exhibit sublinear conditions unless the variation budget grows sublinearly in T [27]. With that in mind, the function variation H_T is defined as

$$H_T = \sum_{t=1}^{T-1} \max_{i\in\mathcal{V}} \max_{\boldsymbol{x}\in\boldsymbol{X}} |f_{i,t+1}(\boldsymbol{x}) - f_{i,t}(\boldsymbol{x})|. \tag{4}$$

This requirement that H_T satisfies sublinear growth in T implies that as the algorithm progresses, the variability in the function difference at the same $\boldsymbol{x}$ decreases over time. Otherwise, irregular variability in the function difference will lead to H_T being at least of the order T because there is always a constant $a > 0$ such that $\max_{i\in\mathcal{V}} \max_{\boldsymbol{x}\in\boldsymbol{X}} |f_{i,t+1}(\boldsymbol{x}) - f_{i,t}(\boldsymbol{x})| > a$ holds. According to [27], this would result in a dynamic regret of order T under any admissible strategy, and the developed algorithm would be unable to achieve long-run-average optimality. Thus, ensuring that the regret is sublinear is crucial, which can be corroborated by the main results presented in Sections 2 and 3.

In distributed online optimization problems defined in (1), projection operations are typically fundamental for addressing constraints, such as distributed online gradient descent in [7], distributed online mirror descent in [10], and distributed online dual averaging in [8]. Generally, the computational burden brought by the projection operations is equivalent to solving a convex quadratic problem [29]. However, in optimization scenarios involving a high-dimensional and structural constraint sets, such as semidefinite programs [30], multiclass classification [31], image reconstruction [32], matrix completion [29, 33], matching pursuit [34], and optimal control [35], the projection step incurs significant computational costs. By contrast, the Frank-Wolfe (FW) algorithm, also known as the projection-free algorithm, offers substantial computational savings [31, 33] owing to the linear oracle $\arg\min_{\boldsymbol{x}\in\boldsymbol{X}} \langle \boldsymbol{x}, \boldsymbol{b}\rangle$, where $\boldsymbol{b}$ is a known vector.

In an early study on centralized online FW optimization [29], Hazan and Kale analyzed the bound of static regret. Following this, numerous studies have explored the FW algorithm within both centralized and distributed OCO frameworks. In the following, we present a brief review of studies most closely related to this paper and make a comparison in detail in Table 1. Zhang et al. [31] earlier developed an FW algorithm paradigm satisfying the distributed online convex optimization framework and obtained the upper bound in $\mathcal{O}(T^{3/4})$ for static regret. Then, Refs. [36, 37] considered several improved variants under a low communication frequency, achieving static regret bounds in $\mathcal{O}(T^{3/4})$ and $\mathcal{O}(T^{2/3}(\log T)^{1/3})$ under conditions of convex and strongly convex loss functions, respectively. Thắng et al. [38] analyzed two algorithm versions of exact and stochastic gradients under smooth loss functions and demonstrated the static regret bound in $\mathcal{O}(\sqrt{T})$. Wang et al. [39] achieved a better regret bound in $\mathcal{O}(T^{2/3})$ than previous distributed online FW methods for general convex cases by exploiting the smoothness of loss functions.

For the more stringent metric, dynamic regret, there has been limited research work on online FW algorithms, especially in distributed scenarios. Wan et al. [40] proposed a centralized online FW algorithm that combines a restarting strategy and analyzed the dynamic regret bound for convex and strongly convex functions, respectively. Kalhan et al. [41] analyzed the dynamic regret of an online FW algorithm under the condition of smooth loss functions and further improved the dynamic regret bounds by incorporating multiple iterations. In distributed scenarios, Zhang et al. [42] developed an online distributed FW optimization algorithm by combining the gradient tracking technique and established the dynamic

Table 1 Comparison among related studies on online Frank-Wolfe optimization

Ref.	Loss function	Problem type	Performance metric (regret)	Linear oracle	Regret bound	Requiring prior knowledge (type)a)
Zhang et al. [31]	Convex	Distributed	Static	$\mathcal{O}(T)$	$\mathcal{O}(T^{3/4})$	Yes (T)
Wan et al. [36]	Convex	Distributed	Static	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(T^{3/4})$	Yes (T)
Wan et al. [37]	Strongly convex	Distributed	Static	$\mathcal{O}(T^{1/3}(\log T)^{2/3})$	$\mathcal{O}(T^{2/3}(\log T)^{1/3})$	Yes (T)
Thắng et al. [38]	Convex	Distributed	Static	$\mathcal{O}(T^{3/2})$	$\mathcal{O}(\sqrt{T})$	Yes (T)
Wan et al. [39]	Convex	Distributed	Static	$\mathcal{O}(T^{2/3})$	$\mathcal{O}(T^{2/3})$	Yes (T)
Kalhan et al. [41]	Convex	Centralized	Dynamic	$\mathcal{O}(T)$ $\mathcal{O}(T^{3/2})$	$\mathcal{O}(\sqrt{T}(1+H_T+\sqrt{D_T}))$ $\mathcal{O}(1+H_T+\sqrt{T})$	Yes (T)
Wan et al. [40]	Convex Strongly convex	Centralized Centralized	Dynamic Dynamic	$\mathcal{O}(T\log_2 T)$ $\mathcal{O}(T\log_2 T)$	$\mathcal{O}(\max\{\sqrt{T}, T^{2/3}H_T^{1/3}\})$ $\mathcal{O}(\max\{\sqrt{TH_T\log T}, \log T\})$	Yes (T)
Zhang et al. [42]	Convex	Distributed	Dynamic	$\mathcal{O}(T)$	$\mathcal{O}(\sqrt{T(1+H_T)}+D_T)$	Yes (H_T and T)
This work	Convex	Distributed	Dynamic	$\mathcal{O}(T^{1+\gamma})$	$\mathcal{O}(T^{1-\gamma}+H_T), 0<\gamma<1$ $\mathcal{O}(1+H_T), 1-\log_T H_T \leqslant \gamma \leqslant 1$	No Yes (H_T and T)

a) To obtain regret results, some input parameters of algorithms, such as step size, may require certain prior knowledge.

regret bound in $\mathcal{O}(\sqrt{T(1+H_T)}+D_T)$ with a non-adaptive step size, where D_T represents the gradient variation. However, achieving this optimal bound depends on two variations of the optimization problem and requires a step size that contains the hard-to-get prior knowledge of H_T, thus resulting in difficulties in accurate tuning of the step size parameter in practice. Following [42], Zhang et al. [43] additionally considered the use of a random quantizer to handle situations with limited communication resources. Lu and Wang [44] investigated and analyzed distributed online FW non-convex optimization algorithms under a new dynamic regret framework.

Motivations and challenges. The above analysis and results in Table 1 reveal the limitations and deficiencies of distributed online FW algorithms in dynamic regret analysis. Is there a method to eliminate these limitations and establish a tighter regret bound?

Multiple iterations, a method that seeks a higher-quality decision by continuously exploiting the function's information at time t, addresses this question. Zhang et al. [45] considered this multiple iteration method for a centralized online gradient descent algorithm and validated that it could significantly enhance the dynamic regret bound. Eshraghi and Liang [46] applied this method to the centralized online mirror descent algorithm. However, both studies required the condition of a strongly convex loss function. Inspired by [41, 42, 45], this paper investigates whether the technique of multiple iterations at round t can remove the dependence of step size on prior knowledge and improve the dynamic regret bound of distributed online FW optimization algorithms under convex loss function.

The idea naturally brings three key challenges in algorithm design and technical analysis.

- Refs. [45,46] explored the multiple iteration method in centralized settings. However, in a distributed setting, it is uncertain how this method is combined with the algorithm.
- Introducing additional inner loops through the multiple iteration method complicates the original technical analysis. Specially, analyzing and proving the consistency error of the algorithm and the convergence relationship between inner and outer loops become more challenging.
- Properly choosing the parameters of the multiple iteration method is crucial to ensure optimal convergence performance.

The main contributions of this work are the following.

(i) Incorporating a multiple iteration technique at each round t, we propose a distributed online multiple FW (DOMFW) algorithm over a time-varying network topology, which efficiently addresses optimization problems with structural and high-dimensional constraint sets. Moreover, based on the gradient tracking method, the global gradient estimation, rather than the gradient of the agent itself, is employed to update the next decision.

(ii) We illustrate that the multiple iteration technique can enhance the dynamic regret bound of the FW algorithm in distributed scenarios. For two inner iteration parameter settings, both dynamic regret bounds $\mathcal{O}(T^{1-\gamma}+H_T)$ with the linear oracle number $\mathcal{O}(T^{1+\gamma})$ are established, where $0<\gamma<1$. Moreover, compared with the existing results in [40–42], this obtained bound does not require a step size dependent on the prior knowledge of H_T, T and can become tighter by tuning the parameter γ.

(iii) With prior knowledge of H_T and T, the optimal bound $\mathcal{O}(1+H_T)$ is obtained, achieving the same regret level as [47], where the latter additionally requires that the loss function is strongly convex and the optima $\boldsymbol{x}_t^* \in \boldsymbol{X}$ satisfies $\nabla F(\boldsymbol{x}_t^*)=\mathbf{0}$. Moreover, we reveal a trade-off between regret bound, computational cost, and communication cost. Finally, the performance of our algorithm is validated and compared through simulations on distributed online ridge regression problems with two constraint sets.

The remaining sections of the paper are structured as follows. Section 2 presents the optimization problem, necessary assumptions, and algorithm design. Section 3 analyzes the convergence results and

Table 2 Description of some common notations

Notation	Definition	Notation	Definition
$\mathbb{R}^n$	n-dimensional Euclidean space	$\mathbb{Z}$	Integer set
$\mathbb{Z}_+$	Positive integer set	$\mathbf{1}$	The vector whose elements are all equal to 1
$\|\boldsymbol{z}\|$	The Euclidean norm of a vector $\boldsymbol{z}$	$\|\boldsymbol{z}\|_1$	1-norm of a vector $\boldsymbol{z}$
$\boldsymbol{g}^{\mathrm{T}}$	Transpose of a vector $\boldsymbol{g}$	$[n]$	The set $\{1,\ldots,n\}$
A_t	Network weighted matrix	$[\boldsymbol{v}]_i$	The i-th element of vector $\boldsymbol{v}$
$[A_t]_{ij}$	The element in the (i,j)th column of A_t	$\mathcal{N}_i^{\mathrm{in}}(t)$	The inner neighbor sets of agent i
$\Phi(\cdot,\cdot),\Phi^K(\cdot,\cdot)$	Transition matrices	K_t	The iteration number of inner loop at time t
$\boldsymbol{x}_{i,t},\boldsymbol{x}_{i,t}^k$	Decision variable and its temporary variant	$\nabla f_{i,t}^k,\overline{\nabla} f_{i,t}^k,\widehat{\nabla} f_{i,t}^k$	Gradient and its temporary variants
M	The diameter of the set $\boldsymbol{X}$	L_X, G_X	Lipschitz continuous and smooth constants of $f_{i,t}$

discussion, while Sections 4 and 5 show the simulation examples and conclusion, respectively.

Notation. Some common symbols used in this paper are stated and summarized in Table 2 for quick reference. Moreover, for two scalar sequences $\{b_i, i\in\mathbb{Z}_+\}$ and $\{c_i>0, i\in\mathbb{Z}_+\}$, $b_i=\mathcal{O}(c_i)$, $b_i=o(c_i)$, and $b_i=\omega(c_i)$ represent that there are scalars a_1,a_2, and a_3 such that $b_i\leqslant a_1c_i$, $b_i<a_1c_i$, and $b_i>a_1c_i$, respectively.

2 Problem formulation

2.1 Optimization problem and some assumptions

In this work, the network information exchange between n agent (nodes) is carried out in a directed time-varying multi-agent graph $\mathcal{G}_t=\{\mathcal{V},\mathcal{E}_t,A_t\}$, where $\mathcal{V}:=\{1,2,\ldots,n\},\mathcal{E}_t\subseteq\mathcal{V}\times\mathcal{V}$, and $A_t\in\mathbb{R}^{n\times n}$ stand for node set, edge set, and weight, respectively. Let $\mathcal{N}_i^{\mathrm{in}}(t)=\{j\mid(j,i)\in\mathcal{E}_t\}\cup\{i\}$ denote inner neighbor sets of agent i, and agent i has permission to receive the information from its neighbor agent in $\mathcal{N}_i^{\mathrm{in}}(t)$ through the network communication. Moreover, when $j\in\mathcal{N}_i^{\mathrm{in}}(t)$, $[A_t]_{ij}>0$ holds, and otherwise $[A_t]_{ij}=0$ holds.

Distributed online optimization problems can be described as an interactive loop with the environment or the adversary as follows:

- Each agent i first commits a decision $\boldsymbol{x}_{i,t}\in\boldsymbol{X}$ with the information reference of its neighbors at every round t;
- The loss $f_{i,t}$ of agent i and its gradient information are revealed by the environment or the adversary;
- Agent i updates the next decision $\boldsymbol{x}_{i,t+1}$ through using the information about loss function $f_{i,t}$.

The goal of this paper is to ensure that the dynamic regret of each agent i achieves sublinear convergence by designing an effective online optimization algorithm, i.e., $\lim_{T\to\infty}(\mathrm{Regret}_d^j(T)/T)=0,\forall j\in\mathcal{V}$.

Throughout the paper, we make the following assumptions.

Assumption 1. (a) When $[A_t]_{ij}>0, t\in[T]$, there exists a positive scalar ζ such that $[A_t]_{ij}>\zeta$. (b) The graph $\mathcal{G}_t$ is strongly connected for all time t. (c) For all $t\in[T]$, A_t is double stochastic, i.e. $\sum_{j=1}^n[A_t]_{ij}=\sum_{i=1}^n[A_t]_{ij}=1$.

Assumption 2. There exists a finite diameter M such that, for any $\boldsymbol{x},\boldsymbol{z}\in\boldsymbol{X}$, $\max_{\boldsymbol{x},\boldsymbol{z}\in\boldsymbol{X}}\|\boldsymbol{x}-\boldsymbol{z}\|\leqslant M$.

Assumption 3. For all $i\in[n]$ and $t\in[T]$, the function $f_{i,t}$ is Lipschitz continuous on constraint set $\boldsymbol{X}$ with a known positive constant L_X, i.e., $|f_{i,t}(\boldsymbol{x})-f_{i,t}(\boldsymbol{z})|\leqslant L_X\|\boldsymbol{x}-\boldsymbol{z}\|,\forall\boldsymbol{x},\boldsymbol{z}\in\boldsymbol{X}$.

Assumption 4. For all $i\in[n]$ and $t\in[T]$, the function $f_{i,t}$ has a Lipschitz gradient on constraint set $\boldsymbol{X}$ with a known positive constant G_X, i.e., $f_{i,t}(\boldsymbol{x})-f_{i,t}(\boldsymbol{z})\leqslant\langle\nabla f_{i,t}(\boldsymbol{z}),\boldsymbol{x}-\boldsymbol{z}\rangle+\frac{G_X}{2}\|\boldsymbol{x}-\boldsymbol{z}\|^2$, $\forall\boldsymbol{x},\boldsymbol{z}\in\boldsymbol{X}$.

Remark 1. Assumptions 2 and 3 are standard in distributed and centralized optimization, and similar settings can be seen in [27,31,33]. It should be pointed out that Assumption 2 can be directly obtained by the properties of the compact set $\boldsymbol{X}$. According to Lemma 2.6 in [1], Assumption 3 implies that the gradient is bounded, i.e., $\|\nabla f_{i,t}(\boldsymbol{x})\|\leqslant L_X$. Assumption 4 is equivalent to the fact that

$$\|\nabla f_{i,t}(\boldsymbol{x})-\nabla f_{i,t}(\boldsymbol{z})\|\leqslant G_X\|\boldsymbol{x}-\boldsymbol{z}\|,\forall\boldsymbol{x},\boldsymbol{z}\in\boldsymbol{X}. \tag{5}$$

2.2 Algorithm DOMFW

In this subsection, we first develop DOMFW, whose description is shown in Algorithm 1. Different from common online algorithm frameworks, Algorithm DOMFW performs multiple iterations at each

Algorithm 1 Distributed online multiple Frank-Wolfe optimization algorithm (DOMFW)

Initialize: Initial variables $\boldsymbol{x}_{i,1} \in \boldsymbol{X}$, parameter $0 < \alpha_t \leqslant 1$, and nondecreasing integer sequence $\{K_t\}$.
for $t = 1, 2, \ldots, T$ **do**
  Set $\boldsymbol{x}^1_{i,t} = \boldsymbol{x}_{i,t}$;
  for $k = 1, 2, \ldots, K_t$ **do**
    for each agent $i \in \mathcal{V}$ **do**
      Agent i receives $\boldsymbol{x}^k_{j,t}$ from $j \in \mathcal{N}^{\rm in}_i(t)$, and updates
$$\hat{\boldsymbol{x}}^k_{i,t} = \textstyle\sum_{j\in\mathcal{N}^{in}_i(t)} [A_t]_{ij}\boldsymbol{x}^k_{j,t}.$$
      Gradient information $\nabla f_{i,t}(\boldsymbol{x})$ is revealed and agent i executes gradient tracking steps:
      if $k = 1$ **then**
        $\overline{\nabla} f^1_{i,t} = \nabla f_{i,t}(\hat{\boldsymbol{x}}^1_{i,t})$;
      else
        $\overline{\nabla} f^k_{i,t} = \widehat{\nabla} f^{k-1}_{i,t} + \nabla f_{i,t}(\hat{\boldsymbol{x}}^k_{i,t}) - \nabla f_{i,t}(\hat{\boldsymbol{x}}^{k-1}_{i,t})$;
      end if
$$\widehat{\nabla} f^k_{i,t} = \textstyle\sum_{j\in\mathcal{N}^{in}_i(t)} [A_t]_{ij}\overline{\nabla} f^k_{j,t};$$
      Frank-Wolfe step: agent i calculates $\boldsymbol{v}^k_{i,t}$ by using a linear oracle and updates $\boldsymbol{x}^{k+1}_{i,t}$:
$$\boldsymbol{v}^k_{i,t} = \arg\min_{\boldsymbol{x}\in\boldsymbol{X}} \langle \boldsymbol{x}, \widehat{\nabla} f^k_{i,t}\rangle;$$
$$\boldsymbol{x}^{k+1}_{i,t} = \hat{\boldsymbol{x}}^k_{i,t} + \alpha_t(\boldsymbol{v}^k_{i,t} - \hat{\boldsymbol{x}}^k_{i,t});$$
    end for
    Agent i updates the decision: $\boldsymbol{x}_{i,t+1} = \boldsymbol{x}^{K_t+1}_{i,t}$ when $k = K_t$ holds;
  end for
end for

time t. In detail, when the decision $\boldsymbol{x}_{i,t}$ of agent i is given, the sequence $\{\boldsymbol{x}^1_{i,t}, \boldsymbol{x}^2_{i,t}, \ldots, \boldsymbol{x}^{K_t}_{i,t}, \boldsymbol{x}^{K_t+1}_{i,t}\}$ is generated. The inner iterations execute from $\boldsymbol{x}^1_{i,t} = \boldsymbol{x}_{i,t}$ and end at step $\boldsymbol{x}_{i,t+1} = \boldsymbol{x}^{K+1}_{i,t}$ after performing the consistency, gradient tracking, and Frank-Wolfe steps in the inner loop.

At time t, the new decision $\boldsymbol{x}_{i,t+1}$ is updated by exploiting the information of the loss function $F_t(\boldsymbol{x})$ multiple times, which is similar to the process of distributed off-line optimization only for the time (round) t. On the one hand, when K_t is set larger, $\boldsymbol{x}_{i,t+1} = \boldsymbol{x}^{K_t+1}_{i,t}$ is actually closer to the optimum $\boldsymbol{x}^*_t \in \arg\min_{\boldsymbol{x}\in\boldsymbol{X}} F_t(\boldsymbol{x})$. On the other hand, Eq. (3) can be converted to the following inequality:

$$\begin{aligned}\mathrm{Regret}^j_d(T) &= \sum_{t=2}^{T}[F_t(\boldsymbol{x}_{j,t}) - F_t(\boldsymbol{x}^*_{t-1}) + F_t(\boldsymbol{x}^*_{t-1}) - F_t(\boldsymbol{x}^*_t)] + F_1(\boldsymbol{x}_{j,1}) - F_1(\boldsymbol{x}^*_1)\\ &= \sum_{t=1}^{T-1}[F_{t+1}(\boldsymbol{x}_{j,t+1}) - F_{t+1}(\boldsymbol{x}^*_t)] + \sum_{t=1}^{T-1}[F_{t+1}(\boldsymbol{x}^*_t) - F_t(\boldsymbol{x}^*_t)] + F_1(\boldsymbol{x}_{j,1}) - F_T(\boldsymbol{x}^*_T)\\ &\leqslant nL_X\sum_{t=1}^{T-1}\|\boldsymbol{x}_{j,t+1} - \boldsymbol{x}^*_t\| + 2nH_T + nL_XM,\end{aligned} \tag{6}$$

where in the last inequality we use Assumption 3 and the fact $F_1(\boldsymbol{x}_{j,1}) - F_T(\boldsymbol{x}^*_T) = F_1(\boldsymbol{x}_{j,1}) - F_1(\boldsymbol{x}^*_T) + \sum_{t=1}^{T-1}[F_t(\boldsymbol{x}^*_T) - F_{t+1}(\boldsymbol{x}^*_T)] \leqslant nM + H_T$.

It is not hard to note that from the perspective of (6), the dynamic regret is related to the term $\|\boldsymbol{x}_{j,t+1} - \boldsymbol{x}^*_t\|$, i.e., the nearness between $\boldsymbol{x}_{j,t+1}$ and $\boldsymbol{x}^*_t$. Thus, from the view of the principle analysis of the multiple iterations method, it is possible to establish a tighter bound than the existing results in Table 1 and remove the dependence of step size on prior knowledge. In particular, compared with the one iteration of the distributed online FW algorithm in [42], algorithm DOMFW executes multiple iterations at each time t to pursue a high-quality decision sequence. From this point, the proposed algorithm can be regarded as an enhanced version of that in [42].

3 Convergence analysis

In this section, we analyze and establish in detail the upper bound of dynamic regret for Algorithm 1. Based on this general result, the effect of the choice of algorithm parameters on convergence is discussed in Corollaries 1 and 2. After that, we further show the advantages of our developed algorithm by comparing with some existing results and reveal a trade-off between obtaining high-quality decisions and saving resources. Before that, some preliminaries are first given.

3.1 Preliminaries

For convenience, we introduce some notations. Firstly, for the square matrix A_t and the positive integer K_t, let $A_t^{K_t}$ denote the K_t-th power of A_t, that is

$$A_t^{K_t} \triangleq \overbrace{A_t \times A_t \times \cdots \times A_t}^{K_t}.$$

It should be noted that $A_t^0 = I$, in which I denotes the identity matrix.

Next, for $t \geqslant s \geqslant 1$, let $\Phi^K(t,s)$ denote the following transition matrix:

$$\Phi(t,s) \triangleq A_t A_{t-1} \ldots A_s, \Phi^K(t,s) \triangleq A_t^{K_t} A_{t-1}^{K_{t-1}} \cdots A_s^{K_s}. \tag{7}$$

In addition, we set $\Phi^K(t,t+1) = I$. It is worth noting that $\Phi^K(t,s)$ defined in (7) is the product of $\sum_{p=s}^{t} K_p$ matrices, and each matrix involved in the product satisfies Assumption 1. By observing this fact and using the convergence property of the transition matrix $\Phi(t,s)$ in Corollary 1 of [14], we obtain the following conditions:

$$\left|[\Phi^K(t,s)]_{ij} - \frac{1}{n}\right| \leqslant \Gamma_1 \sigma_1^{\sum_{p=s}^{t} K_p - 1}, \tag{8}$$

where $\sigma_1 = 1 - \frac{\zeta}{4n^2}$ and $\Gamma_1 = (1 - \frac{\zeta}{4n^2})^{-1}$.

Similarly, note that $\Phi^K(t,s+1)A_s^{K_s - l}$ is the product of $\sum_{p=s}^{t} K_p - l$ matrices satisfying Assumption 1, where $1 \leqslant s \leqslant t$ and $1 \leqslant l \leqslant K_s - 1$. Thus, we have

$$\left|[\Phi^K(t,s+1)A_s^{K_s - l}]_{ij} - \frac{1}{n}\right| \leqslant \Gamma_1 \sigma_1^{\sum_{p=s}^{t} K_p - l - 1}. \tag{9}$$

The conditions in (8) and (9) will be used in the proof of the following equalities.

Moreover, to facilitate the proof and analysis, let the symbols $\boldsymbol{x}_{\mathrm{avg},t}^k$, $\boldsymbol{x}_{\mathrm{avg},t}$, $\boldsymbol{v}_{\mathrm{avg},t}^k$, and $\delta_{i,t}^k$ in (10) denote the running average of $\boldsymbol{x}_{i,t}^k$, the running average of $\boldsymbol{x}_{i,t}$, the running average of $\boldsymbol{v}_{i,t}^k$, and gradient difference of agent i, respectively.

$$\begin{cases} \boldsymbol{x}_{\mathrm{avg},t}^k = \frac{1}{n} \sum\limits_{i=1}^{n} \boldsymbol{x}_{i,t}^k, \boldsymbol{x}_{\mathrm{avg},t} = \frac{1}{n} \sum\limits_{i=1}^{n} \boldsymbol{x}_{i,t}, \\ \boldsymbol{v}_{\mathrm{avg},t}^k = \frac{1}{n} \sum\limits_{i=1}^{n} \boldsymbol{v}_{i,t}^k, \delta_{i,t}^k = \nabla f_{i,t}\left(\hat{\boldsymbol{x}}_{i,t}^{k+1}\right) - \nabla f_{i,t}\left(\hat{\boldsymbol{x}}_{i,t}^{k}\right). \end{cases} \tag{10}$$

Along the above equalities, we can further obtain by combining the relations $\boldsymbol{x}_{i,t} = \boldsymbol{x}_{i,t}^1$ and $\boldsymbol{x}_{i,t+1} = \boldsymbol{x}_{i,t}^{K_t+1}$ from Algorithm 1 that $\boldsymbol{x}_{\mathrm{avg},t}^1 = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_{i,t}^1 = \boldsymbol{x}_{\mathrm{avg},t}$ and $\boldsymbol{x}_{\mathrm{avg},t}^{K_t+1} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_{i,t}^{K_t+1} = \boldsymbol{x}_{\mathrm{avg},t+1}$.

3.2 Main convergence results

Along with the aforementioned basic conditions, some crucial lemmas and the general convergence results of Algorithm 1 are established in this subsection. Lemmas 1 and 2 give the upper bound of the consistency error of the state variable and the upper bound of the tracking error of the estimated gradient, respectively. Different from the analysis of general distributed online optimization [10, 42], the convergence properties of the two new transition matrices described in (8) and (9) play a key role in both technical analyses. Now, we firstly present some necessary lemmas for the regret analysis and the proofs are shown in Appendixes A–C.

Lemma 1 (Consistency error)**.** Suppose Assumptions 1 and 2 hold. Let $\{\boldsymbol{x}_{i,t}\}$ be the decision sequence generated by Algorithm 1. Then, we have for any $T \geqslant 2$, $K_t \geqslant 2$

$$\begin{aligned} \sum_{t=1}^{T}\sum_{i=1}^{n} \|\boldsymbol{x}_{i,t} - \boldsymbol{x}_{\mathrm{avg},t}\| \leqslant{}& \frac{n\Gamma_1}{1 - {\sigma_1}^{K_1}} \sum_{i=1}^{n} \|\boldsymbol{x}_{i,1}\| + \sum_{i=1}^{n} \|\boldsymbol{x}_{i,1} - \boldsymbol{x}_{\mathrm{avg},1}\| \\ &+ \left(\frac{n^2 M \Gamma_1}{\sigma_1\left(1-\sigma_1\right)\left(1-{\sigma_1}^{K_1}\right)} + 2nM\right) \sum_{t=1}^{T} \alpha_t. \end{aligned} \tag{11}$$

Lemma 2 (Tracking error)**.** Suppose Assumptions 1 and 4 hold. Let $\{\widehat{\nabla}^k f_{i,t}\}$ and $\{\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)\}$ be the sequences generated by Algorithm 1. Then, we have for any $T \geqslant 2$, $K_t \geqslant 2$

$$\sum_{t=1}^{T}\sum_{k=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\widehat{\nabla} f_{i,t}^k-\frac{1}{n}\nabla F_t(\boldsymbol{x}_{\mathrm{avg},t}^k)\right\|\leqslant D_1\sum_{t=1}^{T}\alpha_t+D_2\sum_{t=1}^{T}\alpha_t^2K_t, \tag{12}$$

where $D_1=(\frac{2n\Gamma_1 G_X}{1-\sigma_1}+G_X)(nM+\frac{n\Gamma_1}{1-\sigma_1}\sum_{j=1}^n(\|\boldsymbol{x}_{j,1}\|+M))+\frac{n^2\Gamma_1 L_X}{1-\sigma_1}$, $D_2=\frac{2n^2\Gamma_1 G_X M}{1-\sigma_1}(\frac{n\Gamma_1}{1-\sigma_1}+3)+2nMG_X$.

Lemma 3 (Convergence of the inner loop)**.** Suppose Assumptions 2 and 4 hold. Let $\{\boldsymbol{x}_{i,t}\}$ be the decision sequence generated by Algorithm 1. Then, we have at time t

$$\begin{aligned}F_t\left(\boldsymbol{x}_{\mathrm{avg},t}^{k+1}\right)-F_t(x_t^*)&\leqslant(1-\alpha_t)^k\left[F_t\left(\boldsymbol{x}_{\mathrm{avg},t}^1\right)-F_t(x_t^*)\right]\\&\quad+2\alpha_t M\sum_{l=1}^{k}\sum_{i=1}^{n}\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\mathrm{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|+\frac{n\alpha_t G_X M^2}{2}.\end{aligned} \tag{13}$$

In Lemma 3, the dependence of algorithm convergence at time t on the iteration number k of the inner loop is obtained. It should be pointed out that Lemma 3 plays a key role in linking the convergence of the inner loops and the regret analysis of algorithm DOMFW, also called the convergence of outer loops. Based on this, through using the definition of H_T and combining Lemmas 1–3 and the key inequality about the term $(1-\alpha_t)^{K_t}$, we establish the dynamic regret bound of Algorithm 1 in Theorem 1.

Theorem 1. Suppose Assumptions 1–4 hold. Let $\{\boldsymbol{x}_{i,t}\}$ be the decision sequence generated by Algorithm 1 with the parameter $\alpha_t=1/(\rho K_t)$, where $\rho\geqslant 1$ is a constant. Then, for $T\geqslant 2$, $K_t\geqslant 2$, and $j\in\mathcal{V}$, we obtain that

$$\mathrm{Regret}_d^j(T)\leqslant E_1+E_2H_T+E_3\sum_{t=1}^{T}\frac{1}{K_t}, \tag{14}$$

where $E_1=\frac{n^2L_X\Gamma_1}{1-\sigma_1{}^{K_1}}\sum_{i=1}^n\|\boldsymbol{x}_{i,1}\|+nL_X\sum_{i=1}^n\|\boldsymbol{x}_{i,1}-\boldsymbol{x}_{\mathrm{avg},1}\|+nL_XM(1-\mathrm{e}^{-\frac{1}{\rho}})^{-1}$, $E_2=2n(1-\mathrm{e}^{-\frac{1}{\rho}})^{-1}$, and $E_3=2M(\frac{1}{\rho}D_1+\frac{1}{\rho^2}D_2)(1-\mathrm{e}^{-\frac{1}{\rho}})^{-1}+\frac{nG_XM^2}{2\rho}(1-\mathrm{e}^{-\frac{1}{\rho}})^{-1}+\frac{n^2L_XM}{\rho}(\frac{n\Gamma_1}{\sigma_1(1-\sigma_1)(1-\sigma_1{}^{K_1})}+2)$.

Proof. Setting $k=K_t$ in Lemma 3 and using the above relation, we get the following inequality:

$$\begin{aligned}&F_t\left(\boldsymbol{x}_{\mathrm{avg},t+1}\right)-F_t(x_t^*)\\&\leqslant(1-\alpha_t)^{K_t}\left[F_t\left(\boldsymbol{x}_{\mathrm{avg},t}\right)-F_t(x_t^*)\right]+2\alpha_tM\sum_{l=1}^{K_t}\sum_{i=1}^{n}\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\mathrm{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|+\frac{n\alpha_tG_XM^2}{2}.\end{aligned} \tag{15}$$

The term on the left-hand side of the above inequality satisfies $F_t(\boldsymbol{x}_{\mathrm{avg},t+1})-F_t(x_t^*)=F_{t+1}(\boldsymbol{x}_{\mathrm{avg},t+1})-F_{t+1}(\boldsymbol{x}_{t+1}^*)+F_t(\boldsymbol{x}_{\mathrm{avg},t+1})-F_{t+1}(\boldsymbol{x}_{\mathrm{avg},t+1})+F_{t+1}(x_{t+1}^*)-F_t(x_t^*)$. Note that $F_{t+1}(\boldsymbol{x}_{\mathrm{avg},t+1})-F_t(\boldsymbol{x}_{\mathrm{avg},t+1})\leqslant nf_{t+1,\sup}$. Then, we have

$$\begin{aligned}F_{t+1}(\boldsymbol{x}_{\mathrm{avg},t+1})-F_{t+1}(\boldsymbol{x}_{t+1}^*)&\leqslant nf_{t+1,\sup}+F_t(x_t^*)-F_{t+1}(x_{t+1}^*)+(1-\alpha_t)^{K_t}\left[F_t\left(\boldsymbol{x}_{\mathrm{avg},t}\right)-F_t(x_t^*)\right]\\&\quad+2\alpha_tM\sum_{l=1}^{K_t}\sum_{i=1}^{n}\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\mathrm{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|+\frac{n\alpha_tG_XM^2}{2}.\end{aligned} \tag{16}$$

Summing from $t=1$ to $T-1$ on both sides of (16), we get

$$\begin{aligned}&\sum_{t=1}^{T-1}\left[F_{t+1}(\boldsymbol{x}_{\mathrm{avg},t+1})-F_{t+1}(\boldsymbol{x}_{t+1}^*)-F_t(\boldsymbol{x}_{\mathrm{avg},t})+F_t(\boldsymbol{x}_t^*)\right]\\&\quad\leqslant nH_T+F_1(\boldsymbol{x}_1^*)-F_T(\boldsymbol{x}_T^*)+\frac{nG_XM^2}{2}\sum_{t=1}^{T}\alpha_t+2M\sum_{t=1}^{T-1}\sum_{l=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\mathrm{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|\\&\qquad-\sum_{t=1}^{T-1}\left[1-(1-\alpha_t)^{K_t}\right]\left[F_t(\boldsymbol{x}_{\mathrm{avg},t})-F_t(\boldsymbol{x}_t^*)\right].\end{aligned} \tag{17}$$

Note that $\sum_{t=1}^{T-1}[F_{t+1}(\boldsymbol{x}_{\text{avg},t+1})-F_{t+1}(\boldsymbol{x}_{t+1}^*)-F_t(\boldsymbol{x}_{\text{avg},t})+F_t(\boldsymbol{x}_t^*)] = [F_T(\boldsymbol{x}_{\text{avg},T})-F_T(\boldsymbol{x}_T^*)]-[F_1(\boldsymbol{x}_{\text{avg},1})-F_1(\boldsymbol{x}_1^*)]$ and $F_T(\boldsymbol{x}_{\text{avg},T}) - F_T(\boldsymbol{x}_T^*) \geqslant 0$. This, together with (17), implies

$$\begin{aligned}
&\sum_{t=1}^{T}\left[1-(1-\alpha_t)^{K_t}\right]\left[F_t(\boldsymbol{x}_{\text{avg},t})-F_t(\boldsymbol{x}_t^*)\right]\\
&\leqslant nH_T + 2M\sum_{t=1}^{T-1}\sum_{l=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\| + \frac{nG_XM^2}{2}\sum_{t=1}^{T}\alpha_t + F_1(\boldsymbol{x}_{\text{avg},1}) - F_T(\boldsymbol{x}_T^*)\\
&\leqslant 2nH_T + nL_XM + \frac{nG_XM^2}{2}\sum_{t=1}^{T}\alpha_t + 2M\sum_{t=1}^{T-1}\sum_{l=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|, \qquad (18)
\end{aligned}$$

where the last inequality is established based on $F_1(\boldsymbol{x}_{\text{avg},1})-F_T(\boldsymbol{x}_T^*) = F_1(\boldsymbol{x}_{\text{avg},1})-F_1(\boldsymbol{x}_T^*)+\sum_{t=1}^{T-1}[F_t(\boldsymbol{x}_T^*) -F_{t+1}(\boldsymbol{x}_T^*)] \leqslant nL_X\|\boldsymbol{x}_{\text{avg},1}-\boldsymbol{x}_T^*\| + \sum_{t=1}^{T-1}\sum_{i=1}^{n}|f_{i,t}(\boldsymbol{x}_T^*)-f_{i,t+1}(\boldsymbol{x}_T^*)| \leqslant nL_XM + nH_T$.

When α_t is chosen as $1/(\rho K_t)$, in which $\rho \geqslant 1$, we have $(1-\alpha_t)^{\rho K_t} = (1-\frac{1}{\rho K_t})^{\rho K_t} \leqslant \mathrm{e}^{-1}$ where e is the natural constant. Thus, $(1-\alpha_t)^{K_t} \leqslant \mathrm{e}^{-\frac{1}{\rho}}$, which implies $1-(1-\alpha_t)^{K_t} \geqslant 1-\mathrm{e}^{-\frac{1}{\rho}} > 0$. Using this fact, it follows from (18) that

$$\begin{aligned}
&\left(1-\mathrm{e}^{-\frac{1}{\rho}}\right)\sum_{t=1}^{T}[F_t(\boldsymbol{x}_{\text{avg},t})-F_t(\boldsymbol{x}_t^*)]\\
&\leqslant 2nH_T + nL_XM + \frac{nG_XM^2}{2}\sum_{t=1}^{T}\alpha_t + 2M\sum_{t=1}^{T-1}\sum_{l=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|. \qquad (19)
\end{aligned}$$

Based on the above analysis, we are now in position to derive the bound of the regret defined in (3).

$$\begin{aligned}
&\text{Regret}_d^j(T)\\
&\leqslant nL_X\sum_{t=1}^{T}\|\boldsymbol{x}_{j,t}-\boldsymbol{x}_{\text{avg},t}\| + \sum_{t=1}^{T}\left[F_t(\boldsymbol{x}_{\text{avg},t})-F_t(\boldsymbol{x}_t^*)\right]\\
&\leqslant nL_X\sum_{t=1}^{T}\sum_{i=1}^{n}\|\boldsymbol{x}_{i,t}-\boldsymbol{x}_{\text{avg},t}\| + \sum_{t=1}^{T}[F_t(\boldsymbol{x}_{\text{avg},t})-F_t(\boldsymbol{x}_t^*)]\\
&\leqslant \left(1-\mathrm{e}^{-\frac{1}{\rho}}\right)^{-1}\left[2nH_T + nL_XM + \frac{nG_XM^2}{2}\sum_{t=1}^{T}\alpha_t + 2M\sum_{t=1}^{T-1}\sum_{l=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^l\right)-\widehat{\nabla} f_{i,t}^l\right\|\right]\\
&\quad + nL_X\sum_{t=1}^{T}\sum_{i=1}^{n}\|\boldsymbol{x}_{i,t}-\boldsymbol{x}_{\text{avg},t}\|. \qquad (20)
\end{aligned}$$

Then, by applying Lemmas 1 and 2, Eq. (14) can be obtained.

3.3 Discussion

Theorem 1 shows the main results of dynamic regret for algorithm DOMFW. It is easy to note that the regret bound of Algorithm 1 depends on the choices of α_t or sequence $\{K_t\}$. Hence, we have the following corollary by choosing a suitable sequence $\{K_t\}$.

Corollary 1. Let the conditions in Theorem 1 hold. Then, if $H_T = o(T)$ holds, taking $K_t = 1/(\rho\alpha_t) = \lceil\varepsilon_1 t^{\gamma_1}\rceil + 1, \varepsilon_1 > 0, 0 < \gamma_1 < 1, t \in \{1,2,\ldots,T\}$, we have

$$\text{Regret}_d^j(T) \leqslant \mathcal{O}\left(T^{1-\gamma_1} + H_T\right), \boldsymbol{N}_{\text{LO}} \leqslant \mathcal{O}(T^{1+\gamma_1}), \qquad (21)$$

where $\boldsymbol{N}_{\text{LO}}$ denotes the iteration number of linear oracle $\arg\min_{\boldsymbol{x}\in\boldsymbol{X}}\langle\boldsymbol{x},\widehat{\nabla} f_{i,t}^k\rangle$ in Algorithm 1.

In particular, when $1-\log_T H_T \leqslant \gamma_1 < 1, H_T = \omega(1)$, the upper bounds of dynamic regret and $\boldsymbol{N}_{\text{LO}}$ can be establish as $\mathcal{O}(1+H_T)$ and $\mathcal{O}(T^{2-\log_T H_T})$, respectively.

Proof. Substituting the conditions in Corollary 1 into inequality (14), we obtain $\mathrm{Regret}_d^j(T) \leqslant E_1 + E_2 H_T + E_3 \sum_{t=1}^T \frac{1}{\lceil \varepsilon_1 t^{\gamma_1} \rceil + 1} \leqslant E_1 + E_2 H_T + \frac{E_3}{\varepsilon_1} \sum_{t=1}^T \frac{1}{t^{\gamma_1}} \leqslant \mathcal{O}\left(T^{1-\gamma_1} + H_T\right)$ where the last inequality is obtained by using $\sum_{t=1}^T \frac{1}{t^{\gamma_1}} \leqslant 1 + \sum_{t=2}^T \frac{1}{t^{\gamma_1}} \leqslant 1 + \int_1^T \frac{1}{t^{\gamma_1}} \mathrm{d}t \leqslant \frac{1}{1-\gamma_1} T^{1-\gamma_1}$. According to Algorithm 1, $\boldsymbol{N}_{\mathrm{LO}}$ is equivalent to the sum of the inner loop numbers $\sum_{t=1}^T K_t$. Thus, we have $\sum_{t=1}^T K_t \leqslant \int_1^T (\varepsilon_1 t^{\gamma_1} + 2) \mathrm{d}t \leqslant \mathcal{O}(T^{1+\gamma_1})$.

In particular, through substituting the conditions $1 - \log_T H_T \leqslant \gamma_1 < 1, H_T = \omega(1)$ into (21), the bounds of dynamic regret and $\boldsymbol{N}_{\mathrm{LO}}$ are naturally obtained.

Corollary 2. Let the conditions in Theorem 1 hold. Then, if $H_T = o(T)$ holds, taking $K_t = 1/(\rho\alpha_t) = \lceil \varepsilon_2 T^{\gamma_2} \rceil + 1, \varepsilon_2 > 0, 0 < \gamma_2 \leqslant 1, t \in \{1, 2, \dots, T\}$, we have

$$\mathrm{Regret}_d^j(T) \leqslant \mathcal{O}\left(T^{1-\gamma_2} + H_T\right), \boldsymbol{N}_{\mathrm{LO}} \leqslant \mathcal{O}(T^{1+\gamma_2}). \tag{22}$$

In particular, when $1 - \log_T H_T \leqslant \gamma_2 \leqslant 1$, the upper bounds of dynamic regret and $\boldsymbol{N}_{\mathrm{LO}}$ can be establish as $\mathcal{O}(1 + H_T)$ and $\mathcal{O}(T^{2-\log_T H_T})$, respectively.

Proof. Similar to the proof of Corollary 1, we have $\mathrm{Regret}_d^j(T) \leqslant E_1 + E_2 H_T + \frac{E_3}{\varepsilon_2} T^{1-\gamma_2}$, and $\sum_{t=1}^T K_t \leqslant \mathcal{O}(T^{1+\gamma_2})$.

Remark 2. Specially, the regret bounds in Corollaries 1 and 2 match the centralized results in [41] and the distributed results in [42], and have more significant advantages than them.

- Compared with the bound $\mathcal{O}(\sqrt{T}(1 + H_T + \sqrt{D_T}))$ of the first algorithm in [41], our results are less conservative and tighter, and can remove the dependency on D_T, where D_T denotes gradient variation. For example, the bound $\mathcal{O}(\sqrt{T} + H_T)$ is better when $\gamma_1 = 0.5$ holds. The range of H_T in this paper is further enhanced to $\mathcal{O}(T)$ instead of $\mathcal{O}(\sqrt{T})$ if a sublinear regret bound is expected, which effectively expands the application field of optimization problems.
- In contrast to the bound $\mathcal{O}(1 + \sqrt{T} + H_T)$ of the second algorithm in [41], the regret bounds in Corollaries 1 and 2 are more flexible and can achieve a better convergence performance, i.e., the cases when $\gamma_1 \in (0.5, 1)$ or $\gamma_2 \in (0.5, 1]$. Moreover, the step size in Corollary 1 does not require the prior knowledge of T.
- In the distributed work [42], the establishment of the optimal dynamic regret bound $\mathcal{O}(\sqrt{T(1 + H_T)} + D_T)$ depends on D_T and a step size with knowledge of H_T, which leads to difficulties in accurate tuning of the step size parameter in practice. In contrast, our results remove the aforementioned limitations and achieve a tighter regret bound, such as the bound $\mathcal{O}(\sqrt{T} + H_T)$ when $\gamma_1 = 0.5$, than [42] under the same conditions of loss function.

Remark 3 (Optimal bound)**.** Corollaries 1 and 2 reveal that under the prior knowledge of H_T and T, the general bounds in (21) and (22) can be improved to the optimal bound $\mathcal{O}(1 + H_T)$, which also can achieve a saving in computing and communication resources by a tunning of lower inner loop number K_t. In particular, this optimal bound $\mathcal{O}(1 + H_T)$ is the same as the regret level in [47], where the latter requires that the loss function is strongly convex and the optima $\boldsymbol{x}_t^* \in \boldsymbol{X}$ satisfies $\nabla F(\boldsymbol{x}_t^*) = \boldsymbol{0}$. Therefore, this also reflects that H_T has a large impact on dynamic regret. Moreover, in practice, the order of H_T over time T may be difficult to determine exactly, and its estimation $\hat{H}_T := \mathcal{O}(T^{\gamma_3}) \geqslant H_T$ based on factitious experience is undoubtedly a practical substitute, where γ_3 is a known constant.

Remark 4 (Communication number)**.** It is not hard to note that from Algorithm 1, agent i communicates $2K_t$ times with its neighbor at time t. As the algorithm runs over time T, the level of communication number attaches $\mathcal{O}(T^{1+\gamma_1})$ under the parameter settings of Corollary 1, which is a weakness of this paper since more communication resources are used than non-multiple algorithms. At the time t, the multiple communications between agents are to ensure that the algorithm can obtain detailed neighbor information to output a more reliable decision $\boldsymbol{x}_{i,t+1}$. Thus, from this point of view, it is reasonable.

3.4 Trade-off between regret bound, computational cost, and communication cost

From Corollaries 1 and 2, it can be known that the choice of $\gamma_1 \in (0, 1)$ (or $\gamma_2 \in (0, 1]$), i.e., the setting of K_t, is closely linked to regret bound, computational and communication costs, which further implies a trade-off between them. Specifically, to establish the desired regret bound in $\mathcal{O}(T^{1-\gamma_1} + H_T)$, Algorithm 1 needs to perform the calculations with the scale $\mathcal{O}(T^{1+\gamma_1})$, including gradient solvers and linear oracles, as well as perform the communication between agents with the scale $\mathcal{O}(T^{1+\gamma_1})$. Intuitively, Table 3 shows a more nuanced discussion on how these factors interplay in the general distributed OCO problems. From

Table 3 Relationship between the parameter γ_1 (or γ_2), the regret bound, $\boldsymbol{N}_{\rm LO}$, and communication number

$H_T(\hat{H}_T)$	γ_1 (or γ_2)	$\boldsymbol{N}_{\rm LO}$	Communication number	$\mathrm{Regret}_d^j(T)$
Unknown	0.3	$\mathcal{O}(T^{1.3})$	$\mathcal{O}(T^{1.3})$	$\mathcal{O}(T^{0.7}+H_T)$
	0.5	$\mathcal{O}(T^{1.5})$	$\mathcal{O}(T^{1.5})$	$\mathcal{O}(\sqrt{T}+H_T)$
	0.7	$\mathcal{O}(T^{1.7})$	$\mathcal{O}(T^{1.7})$	$\mathcal{O}(T^{0.3}+H_T)$
	$\gamma_2=1$	$\mathcal{O}(T^{2})$	$\mathcal{O}(T^{2})$	$\mathcal{O}(1+H_T)$
$\mathcal{O}(\sqrt{T})$	0.3	$\mathcal{O}(T^{1.3})$	$\mathcal{O}(T^{1.3})$	$\mathcal{O}(T^{0.7})$
	0.5	$\mathcal{O}(T^{1.5})$	$\mathcal{O}(T^{1.5})$	$\mathcal{O}(\sqrt{T})$
	0.7	$\mathcal{O}(T^{1.7})$	$\mathcal{O}(T^{1.7})$	$\mathcal{O}(1+H_T)$
	$\gamma_2=1$	$\mathcal{O}(T^{2})$	$\mathcal{O}(T^{2})$	$\mathcal{O}(1+H_T)$

Table 3, it is not hard to note that a larger parameter γ_1 (or γ_2) produces a tighter regret bound while also incurring greater resource costs from $\boldsymbol{N}_{\rm LO}$ and communication number. In other words, more investment in computational and communication costs can help to tighten the regret bound.

In particular, for a class of distributed OCO problems with the limited communication and computational resources, the resource costs caused by the adjustment of γ_1 (or γ_2) dominate this trade-off and the desired regret bound may be difficult to achieve. Thus, the optimal γ_1 (or γ_2) should be chosen within the computational and communication cost constraints. Moreover, it can be observed in Table 3 that blindly pursuing a tight regret bound may deviate from our original expectations, such as the cases that $H_T(\hat{H}_T)=\mathcal{O}(\sqrt{T}), \gamma_1=0.7, \gamma_2=0.7, 1$. In this case, the resource burden caused by excessively large parameters has a greater impact than the performance benefit. Therefore, considering the computational and communication costs, the parameter γ_1 (or γ_2) that satisfies $\mathcal{O}(T^{1-\gamma_1})=\mathcal{O}(H_T)$ (or $\mathcal{O}(T^{1-\gamma_2})=\mathcal{O}(H_T)$) or its nearby value may be one of the trade-off solutions in practical applications.

4 Simulation

The distributed ridge regression problem is investigated in this section to validate the performance of the proposed algorithm shown as

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}}\ \sum_{t=1}^{T}\sum_{i=1}^{n}\left[\frac{1}{2}\left(\boldsymbol{a}_{i,t}^{\rm T}\boldsymbol{x}-b_{i,t}\right)^2+\lambda_1\|\boldsymbol{x}\|_2^2\right],\\ &\text{s.t. } \boldsymbol{x}\in\boldsymbol{X}, \end{aligned} \tag{23}$$

where $\boldsymbol{a}_{i,t}\in\mathbb{R}^d$ is the feature vector and generated randomly and uniformly in $[-5,5]^d$, and $b_{i,t}\in\mathbb{R}$ represents the label information. The label $b_{i,t}$ satisfies $b_{i,t}=\boldsymbol{a}_{i,t}^{\rm T}\boldsymbol{x}_0+\frac{\zeta_{i,t}}{4t}$ where $\zeta_{i,t}$ is generated randomly in $[0,1]$. In this simulation, we consider the following two constraints: (i) the unit simplex constraint $\boldsymbol{X}:=\{\boldsymbol{x}|\mathbf{1}^{\rm T}\boldsymbol{x}=1\},[\boldsymbol{x}]_i\geqslant 0$; (ii) the norm ball constraint $\boldsymbol{X}:=\{\boldsymbol{x}|\|\boldsymbol{x}\|_1\leqslant 2\}$. To intuitively verify the convergence performance of the developed algorithm, we define the global average dynamic regret (ADR) $\frac{1}{n}\sum_{j=1}^{n}[\mathrm{Regret}_d^j(T)/T]$, the upper envelope $\sup_j\{\mathrm{Regret}_d^j(T)/T\}$, and the lower envelope $\inf_j\{\mathrm{Regret}_d^j(T)/T\}$ of ADR, respectively. In the following cases, we set $n=20$ and $\lambda_1=5\times10^{-6}$.

4.1 Unit simplex constraint

Under the unit simplex constraint, the convergence performance of Algorithm 1 is firstly analyzed. In Figure 1(a), the simulation results show that the upper envelope, the lower envelope, and the global value of $\mathrm{Regret}_d^j(T)/T$ are convergent for Algorithm 1 under the condition $d=8, K_t=\lceil 2\sqrt{t}\rceil+1, \rho=3$, which is consistent with our theoretical results. Next, we compare the convergence performance of Algorithm 1 with the time-varying parameter K_t, with fixed parameter K_T and distributed online Frank-Wolfe (DOFW) algorithm in [42], where the related parameters are set as $K_t=\lceil 2\sqrt{t}\rceil+1, \rho=3$, $K_T=\lceil 2\sqrt{T}\rceil+1, \rho=3$, and $\alpha=1/(4T^{0.4})$ [42], respectively. Theoretically, according to Theorem 1 of [42], algorithm DOFW achieves the practical regret bounds in order $\mathcal{O}(T^{0.6}+T^{0.4}H_T+D_T)$. Based on Remark 2(iii), $\mathcal{O}(\sqrt{T})\leqslant\mathcal{O}(T^{0.6})$, and $\mathcal{O}(H_T)\leqslant\mathcal{O}(T^{0.4}H_T)$, it is obtained that $\mathcal{O}(\sqrt{T}+H_T)\leqslant\mathcal{O}(T^{0.6}+T^{0.4}H_T+D_T)$, which implies that algorithm DOMFW converges faster than algorithm DOFW. By observing Figure 1(b), this simulation result is consistent with the above theoretical analysis. Moreover, under this constraint, the algorithm with fixed parameter K_T performs better than that with K_t.

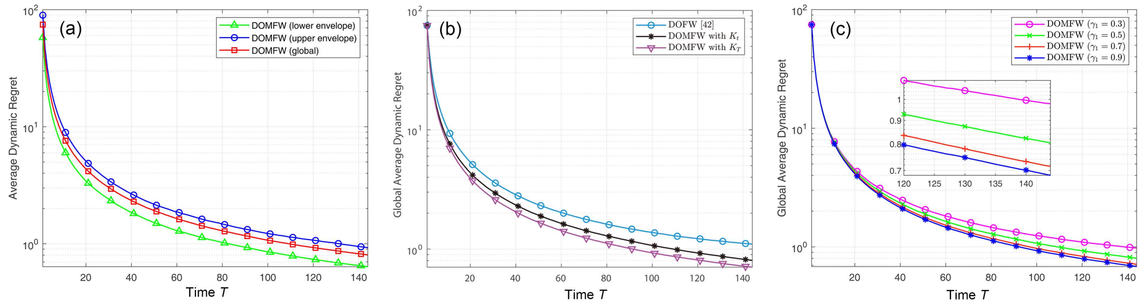

Figure 1 (Color online) Convergence performance under the unit simplex constraint. (a) Three ADRs of Algorithm 1; (b) the comparisons with the existing algorithms; (c) the comparisons under different inner number K_t.

Now, we study the effect of inner iteration number K_t on the convergence performance of Algorithm 1 under the parameters $\rho = 3, \varepsilon_1 = 2$. The plots in Figure 1(c) show the global ADRs for four different settings of K_t in the same network, i.e., the cases when $\gamma_1 = 0.3, 0.5, 0.7, 0.9$. The simulation results clearly reveal that the convergence performance of Algorithm 1 is getting better and better as K_t increases, which corresponds to the results in Corollary 1. Meanwhile, when the parameter γ_1 changes from 0.7 to 0.9, the improvement of the convergence effect becomes slow. From $b_{i,t}$, the estimation $\hat{H}_T$ can be approximated as $\mathcal{O}(\ln T)$. Thus, as γ_1 gets larger, especially for $\gamma_1 \in [0.7, 1)$, the effect of H_T in the upper bound of $\mathrm{Regret}_d^j(T)$ is more emphasized and the performance improvement brought by γ_1 is weaker.

4.2 Norm ball constraint

In this subsection, we study the convergence performance of Algorithm 1 under the norm ball constraint with the dimension $d = 16$. Set $\varepsilon_1 = \varepsilon_2 = 0.3$, $\gamma_1 = \gamma_2 = 0.5, \rho = 3.5$. Similarly, by observing three regret curves in Figure 2(a), we obtain that Algorithm 1 is convergent for this constraint. Next, to verify the effect of the multiple iteration method, we compared algorithm DOFW in [42] and Algorithm 1 under two inner iteration parameters. From Figure 2(b), algorithm DOMFW has better convergence performance than that without multiple iterations [42], which corresponds to the theoretical justification $\mathcal{O}(\sqrt{T} + H_T) \leqslant \mathcal{O}(T^{0.6} + T^{0.4}H_T + D_T)$ and reflects the significant advantages of the multiple iteration method. For two parameter settings K_t and K_T, the performance generated from the latter is better, while the former does not depend on prior knowledge of T.

Further, taking K_t as an example, we explore the convergence performance of Algorithm 1 under four different settings of γ_1, i.e., $0.3, 0.5, 0.7, 0.9$ and the settings $\rho = 3.5, \varepsilon_1 = 0.3$. Figure 2(c) clearly shows that as the order setting of K_t over time t increases, the global ADR of Algorithm 1 is getting better. Similar to Figure 1(c), when γ_1 changes from 0.7 to 0.9, the improvement of the convergence effect is weak while the increase in the computational cost and communication cost is great. Therefore, as described in Section 3, a proper setting of parameters ε_1(or ε_2) and γ_1(or γ_2) is very important to trade off the relationship between obtaining high-quality decision and saving resources in practical applications.

Finally, with the goal of further underscoring the effectiveness and broad applicability of the developed algorithm, we implement the algorithm DOMFW on a real-world dataset. From the LIBSVM[1)] repository, we select the abalone dataset with 8 features and 4177 samples. Under the same constraint and the settings $\rho = 12, \varepsilon_1 = 0.25, \gamma_1 = 0.5$, it can be observed from Figure 3(a) that algorithm DOMFW is convergent. Further, we carry out a comparison with algorithm DOFW [42]. The plots in Figure 3(b) show that algorithm DOMFW using the settings K_t and K_T converges faster than the one in [42]. Moreover, from Figure 3(c), it is known that as γ_1 increases, the better convergence performance of Algorithm 1 can be achieved, which corresponds to the case on synthetic data and the theoretical analysis in Corollary 1.

5 Conclusion

In this work, we have developed the distributed online multiple FW algorithm for solving distributed online optimization problems over a time-varying multi-agent network. Leveraging a projection-free operation, the proposed algorithm significantly reduces computational costs (at one step), especially in

1) https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

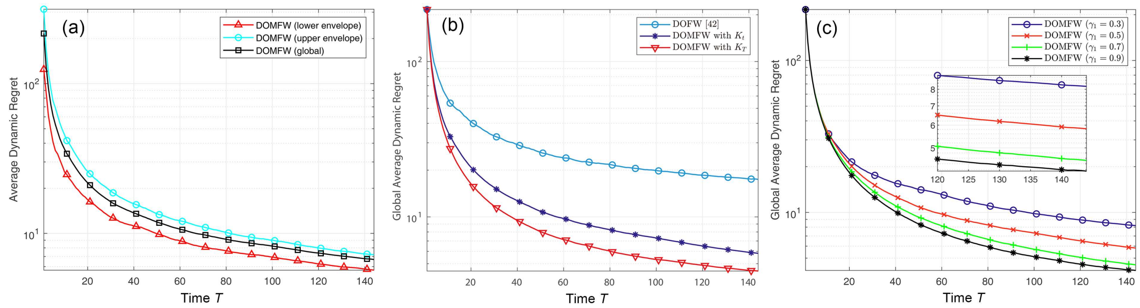

Figure 2 (Color online) Convergence performance for the norm ball constraint on synthetic data. (a) Three ADRs of Algorithm 1; (b) the comparisons with the existing algorithms; (c) the comparisons under different inner number K_t.

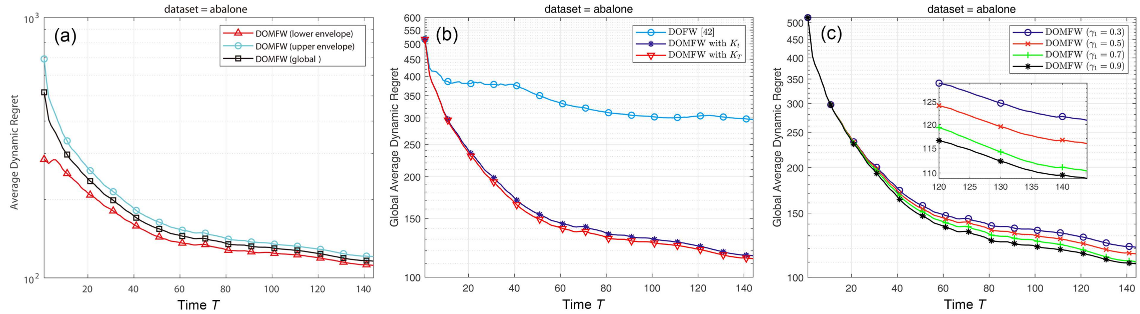

Figure 3 (Color online) Convergence performance for the norm ball constraint on abalone dataset. (a) Three ADRs of Algorithm 1; (b) the comparisons with the existing algorithms; (c) the comparisons under different inner number K_t.

optimization scenarios involving high-dimensional and structural constraint sets. Moreover, we have confirmed that the multiple iteration technique can enhance the dynamic regret bound of the FW algorithm in distributed scenarios. We have established the regret bound $\mathcal{O}(T^{1-\gamma} + H_T), 0 < \gamma < 1$ with the linear oracle number $\mathcal{O}(T^{1+\gamma})$. This bound is tighter than those obtained without inner iteration loops and does not require a step size dependent on the prior knowledge of H_T, T. In particular, when the order or estimated order of H_T over time T is available, the optimal bound $\mathcal{O}(1 + H_T)$ can be obtained. Moreover, we have revealed a trade-off between dynamic regret bound, computational cost, and communication cost. The theoretical results have been validated through simulations on distributed online ridge regression problems with two constraint sets.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62273181, 62373190, 62221004) and in part by Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant No. KYCX22_0453).

References

1 Shalev-Shwartz S. Online learning and online convex optimization. FNT Machine Learn, 2011, 4: 107–194
2 Hazan E. Introduction to online convex optimization. FNT Optimization, 2015, 2: 157–325
3 Li X X, Xie L H, Li N. A survey on distributed online optimization and online games. Annu Rev Control, 2023, 56: 100904
4 Yuan D M, Hong Y G, Ho D W C, et al. Distributed mirror descent for online composite optimization. IEEE Trans Automat Contr, 2021, 66: 714–729
5 Yi X L, Li X X, Yang T, et al. Regret and cumulative constraint violation analysis for distributed online constrained convex optimization. IEEE Trans Automat Contr, 2023, 68: 2875–2890
6 Wang C, Xu S Y. Distributed online constrained optimization with feedback delays. IEEE Trans Neural Netw Learn Syst, 2024, 35: 1708–1720
7 Yan F, Sundaram S, Vishwanathan S V N, et al. Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties. IEEE Trans Knowl Data Eng, 2013, 25: 2483–2493
8 Hosseini S, Chapman A, Mesbahi M. Online distributed optimization via dual averaging. In: Proceedings of the 52nd IEEE Conference on Decision and Control, 2013. 1484–1489
9 Zhang W T, Shi Y, Zhang B Y, et al. Dynamic regret of distributed online saddle point problem. IEEE Trans Automat Contr, 2024, 69: 2522–2529
10 Shahrampour S, Jadbabaie A. Distributed online optimization in dynamic environments using mirror descent. IEEE Trans Automat Contr, 2017, 63: 714–725
11 Wang Y H, Zeng X L, Zhao W X, et al. A zeroth-order algorithm for distributed optimization with stochastic stripe observations. Sci China Inf Sci, 2023, 66: 199202
12 Gao J, Liu X-W, Dai Y-H, et al. Achieving geometric convergence for distributed optimization with Barzilai-Borwein step sizes. Sci China Inf Sci, 2022, 65: 149204
13 Chen J, Kai S X. Cooperative transportation control of multiple mobile manipulators through distributed optimization. Sci China Inf Sci, 2018, 61: 120201

14 Nedić A, Olshevsky A, Ozdaglar A, et al. Distributed subgradient methods and quantization effects. In: Proceedings of the 47th IEEE Conference on Decision and Control, 2008. 4177–4184
15 Nedić A, Liu J. Distributed optimization for control. Annu Rev Control Robot Auton Syst, 2018, 1: 77–103
16 Yang T, Yi X L, Wu J F, et al. A survey of distributed optimization. Annu Rev Control, 2019, 47: 278–305
17 Zhang J Q, You K Y, Xie L H. Innovation compression for communication-efficient distributed optimization with linear convergence. IEEE Trans Automat Contr, 2023, 68: 6899–6906
18 Xiong M H, Zhang B Y, Yuan D M, et al. Distributed quantized mirror descent for strongly convex optimization over time-varying directed graph. Sci China Inf Sci, 2022, 65: 202202
19 Li Z H, Ding Z T. Distributed optimization on unbalanced graphs via continuous-time methods. Sci China Inf Sci, 2018, 61: 129204
20 Yu W W, Li C J, Yu X H, et al. Economic power dispatch in smart grids: a framework for distributed optimization and consensus dynamics. Sci China Inf Sci, 2018, 61: 012204
21 Xie S Y, Wang L Y, Nazari M H, et al. Distributed optimization with Markovian switching targets and stochastic observation noises with applications to DC microgrids. Sci China Inf Sci, 2022, 65: 222205
22 Cheng S S, Lei J L, Zeng X L, et al. Effective distributed algorithm for solving linear matrix equations. Sci China Inf Sci, 2023, 66: 189202
23 Liu C X, Li H P, Shi Y. A unitary distributed subgradient method for multi-agent optimization with different coupling sources. Automatica, 2020, 114: 108834
24 Li W J, Zeng X L, Hong Y G, et al. Distributed consensus-based solver for semi-definite programming: an optimization viewpoint. Automatica, 2021, 131: 109737
25 Xu J M, Soh Y C. A distributed simultaneous perturbation approach for large-scale dynamic optimization problems. Automatica, 2016, 72: 194–204
26 Zinkevich M. Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of the 20th International Conference on Machine Learning, 2003. 928–936
27 Besbes O, Gur Y, Zeevi A. Non-stationary stochastic optimization. Operations Res, 2015, 63: 1227–1244
28 Xu Z R, Zhou H Y, Tzoumas V. Online submodular coordination with bounded tracking regret: theory, algorithm, and applications to multi-robot coordination. IEEE Robot Autom Lett, 2023, 8: 2261–2268
29 Hazan E, Kale S. Projection-free online learning. In: Proceedings of the 29th International Conference on International Conference on Machine Learning, 2012. 1843–1850
30 Hazan E. Sparse approximate solutions to semidefinite programs. In: Proceedings of Latin American Symposium on Theoretical Informatics, 2008. 306–316
31 Zhang W P, Zhao P L, Zhu W W, et al. Projection-free distributed online learning in networks. In: Proceedings of the 34th International Conference on Machine Learning, 2017. 4054–4062
32 Harchaoui Z, Juditsky A, Nemirovski A. Conditional gradient algorithms for norm-regularized smooth convex optimization. Math Program, 2015, 152: 75–112
33 Wai H T, Lafond J, Scaglione A, et al. Decentralized Frank-Wolfe algorithm for convex and nonconvex problems. IEEE Trans Automat Contr, 2017, 62: 5522–5537
34 Locatello F, Khanna R, Tschannen M, et al. A unified optimization view on generalized matching pursuit and Frank-Wolfe. In: Proceedings of Artificial Intelligence and Statistics, 2017. 860–868
35 Wu Z S, Teo K L. A conditional gradient method for an optimal control problem involving a class of nonlinear second-order hyperbolic partial differential equations. J Math Anal Appl, 1983, 91: 376–393
36 Wan Y Y, Tu W W, Zhang L J. Projection-free distributed online convex optimization with $O(\sqrt{T})$ communication complexity. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 9818–9828
37 Wan Y Y, Wang G H, Tu W W, et al. Projection-free distributed online learning with sublinear communication complexity. J Mach Learn Res, 2022, 23: 7742–7794
38 Thắng N K, Srivastav A, Trystram D, et al. A stochastic conditional gradient algorithm for decentralized online convex optimization. J Parallel Distr Comput, 2022, 169: 334–351
39 Wang Y B, Wan Y Y, Zhang S M, et al. Distributed projection-free online learning for smooth and convex losses. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 10226–10234
40 Wan Y Y, Xue B, Zhang L J. Projection-free online learning in dynamic environments. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 10067–10075
41 Kalhan D S, Bedi A S, Koppel A, et al. Dynamic online learning via Frank-Wolfe algorithm. IEEE Trans Signal Process, 2021, 69: 932–947
42 Zhang W T, Shi Y, Zhang B Y, et al. Dynamic regret of distributed online Frank-Wolfe convex optimization. 2023. ArXiv:2302.00663
43 Zhang W T, Shi Y, Zhang B Y, et al. Quantized distributed online projection-free convex optimization. IEEE Control Syst Lett, 2023, 7: 1837–1842
44 Lu K H, Wang L. Online distributed optimization with nonconvex objective functions via dynamic regrets. IEEE Trans Automat Contr, 2023, 68: 6509–6524
45 Zhang L J, Yang T B, Yi J F, et al. Improved dynamic regret for non-degenerate functions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017. 732–741
46 Eshraghi N, Liang B. Dynamic regret bounds without Lipschitz continuity: online convex optimization with multiple mirror descent steps. In: Proceedings of American Control Conference (ACC), 2022. 228–235
47 Wan Y Y, Zhang L J, Song M L. Improved dynamic regret for online Frank-Wolfe. In: Proceedings of the 36th Annual Conference on Learning Theory, 2023. 3304–3327

Appendix A Proof of Lemma 1

According to Algorithm 1, letting $A_t^{K_t}$ denote the product of K_t multiplications of A_t, we get

$$\boldsymbol{x}_{i,t+1} = \sum_{j=1}^{n}[A_t]_{ij}\boldsymbol{x}_{j,t}^{K_t} + \alpha_t\left(\boldsymbol{v}_{i,t}^{K_t} - \hat{\boldsymbol{x}}_{i,t}^{K_t}\right)$$

$$= \sum_{j=1}^{n}[A_t^{K_t}]_{ij}\boldsymbol{x}_{j,t} + \alpha_t \sum_{l=1}^{K_t-1}\sum_{j=1}^{n}[A_t^{K_t-l}]_{ij}\left(\boldsymbol{v}_{j,t}^{l} - \hat{\boldsymbol{x}}_{j,t}^{l}\right) + \alpha_t\left(\boldsymbol{v}_{i,t}^{K_t} - \hat{\boldsymbol{x}}_{i,t}^{K_t}\right)$$
$$= \sum_{j=1}^{n}[\Phi^K(t,1)]_{ij}\boldsymbol{x}_{j,1} + \alpha_t\left(\boldsymbol{v}_{i,t}^{K_t} - \hat{\boldsymbol{x}}_{i,t}^{K_t}\right) + \sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\sum_{j=1}^{n}\alpha_s[\Phi^K(t,s+1)A_s^{K_s-l}]_{ij}\left(\boldsymbol{v}_{j,s}^{l} - \hat{\boldsymbol{x}}_{j,s}^{l}\right)$$
$$+ \sum_{s=1}^{t-1}\sum_{j=1}^{n}\alpha_s[\Phi^K(t,s+1)]_{ij}\left(\boldsymbol{v}_{j,s}^{K_s} - \hat{\boldsymbol{x}}_{j,s}^{K_s}\right). \tag{A1}$$

According to Algorithm 1, we have

$$\boldsymbol{x}_{\mathrm{avg},t}^{K_t+1} = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{n}[A_t]_{ij}\boldsymbol{x}_{j,t}^{K_t} + \alpha_t\left(\boldsymbol{v}_{i,t}^{K_t} - \sum_{j=1}^{n}[A_t]_{ij}\boldsymbol{x}_{j,t}^{K_t}\right)\right]$$
$$= \boldsymbol{x}_{\mathrm{avg},t}^{K_t} + \alpha_t(\boldsymbol{v}_{\mathrm{avg},t}^{K_t} - \boldsymbol{x}_{\mathrm{avg},t}^{K_t}). \tag{A2}$$

By iteration with respect to K_t, we further obtain $\boldsymbol{x}_{\mathrm{avg},t}^{K_t+1} = \boldsymbol{x}_{\mathrm{avg},t} + \alpha_t\sum_{l=1}^{K_t}(\boldsymbol{v}_{\mathrm{avg},t}^{l} - \boldsymbol{x}_{\mathrm{avg},t}^{l})$. Note that $\boldsymbol{x}_{\mathrm{avg},t+1} = \boldsymbol{x}_{\mathrm{avg},t}^{K_t+1}$. Therefore, we have

$$\boldsymbol{x}_{\mathrm{avg},t+1} = \boldsymbol{x}_{\mathrm{avg},1} + \sum_{s=1}^{t}\sum_{l=1}^{K_s}\alpha_s(\boldsymbol{v}_{\mathrm{avg},s}^{l} - \boldsymbol{x}_{\mathrm{avg},s}^{l})$$
$$= \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}_{j,1} + \frac{1}{n}\sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\sum_{j=1}^{n}\alpha_s(\boldsymbol{v}_{j,s}^{l} - \hat{\boldsymbol{x}}_{j,s}^{l}) + \frac{1}{n}\sum_{s=1}^{t-1}\sum_{j=1}^{n}\alpha_s(\boldsymbol{v}_{j,s}^{K_s} - \hat{\boldsymbol{x}}_{j,s}^{K_s}) + \frac{\alpha_t}{n}\sum_{j=1}^{n}(\boldsymbol{v}_{j,t}^{K_t} - \hat{\boldsymbol{x}}_{j,t}^{K_t}). \tag{A3}$$

By combining (A1) and (A3), it follows that

$$\|\boldsymbol{x}_{i,t+1} - \boldsymbol{x}_{\mathrm{avg},t+1}\|$$
$$\leqslant \sum_{j=1}^{n}\left|[\Phi^K(t,1)]_{ij} - \frac{1}{n}\right|\|\boldsymbol{x}_{j,1}\| + \alpha_t\left\|\boldsymbol{v}_{i,t}^{K_t} - \hat{\boldsymbol{x}}_{i,t}^{K_t}\right\| + \sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\sum_{j=1}^{n}\alpha_s\left|[\Phi^K(t,s+1)A_s^{K_s-l}]_{ij} - \frac{1}{n}\right|\left\|\boldsymbol{v}_{j,s}^{l} - \hat{\boldsymbol{x}}_{j,s}^{l}\right\|$$
$$+ \sum_{s=1}^{t-1}\sum_{j=1}^{n}\alpha_s\left|[\Phi^K(t,s+1)]_{ij} - \frac{1}{n}\right|\left\|\boldsymbol{v}_{j,s}^{K_s} - \hat{\boldsymbol{x}}_{j,s}^{K_s}\right\| + \frac{\alpha_t}{n}\sum_{j=1}^{n}\left\|\boldsymbol{v}_{j,t}^{K_t} - \hat{\boldsymbol{x}}_{j,t}^{K_t}\right\|. \tag{A4}$$

Note that, for any appropriate numbers j, s, l, the inequality $\|\boldsymbol{v}_{j,s}^{l} - \hat{\boldsymbol{x}}_{j,s}^{l}\| \leqslant M$ holds. Then, through recalling (8), (9) and using the condition, Eq. (A4) follows

$$\|\boldsymbol{x}_{i,t+1} - \boldsymbol{x}_{\mathrm{avg},t+1}\|$$
$$\leqslant \Gamma_1\sigma_1^{\sum_{p=1}^{t}K_p-1}\sum_{j=1}^{n}\|\boldsymbol{x}_{j,1}\| + nM\Gamma_1\sum_{s=1}^{t-1}\alpha_s\sigma_1^{\sum_{p=s+1}^{t}K_p-1} + nM\Gamma_1\sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\alpha_s\sigma_1^{\sum_{p=s}^{t}K_p-l-1} + 2\alpha_t M$$
$$\leqslant \Gamma_1\sigma_1^{tK_1-1}\sum_{j=1}^{n}\|\boldsymbol{x}_{j,1}\| + nM\Gamma_1\sum_{s=1}^{t}\alpha_s\sigma_1^{(t-s)K_1-1} + nM\Gamma_1\sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\alpha_s\sigma_1^{(t-s)K_1+K_t-l-1} + 2\alpha_t M, \tag{A5}$$

where in the last inequality we use the non-decreasing property of the sequence $\{K_t\}$. Based on (A5), we can further obtain

$$\sum_{t=1}^{T}\sum_{i=1}^{n}\|\boldsymbol{x}_{i,t} - \boldsymbol{x}_{\mathrm{avg},t}\| = \sum_{t=1}^{T-1}\sum_{i=1}^{n}\|\boldsymbol{x}_{i,t+1} - \boldsymbol{x}_{\mathrm{avg},t+1}\| + \sum_{i=1}^{n}\|\boldsymbol{x}_{i,1} - \boldsymbol{x}_{\mathrm{avg},1}\|$$
$$\leqslant n\Gamma_1\sum_{t=1}^{T-1}\sigma_1^{tK_1-1}\sum_{j=1}^{n}\|\boldsymbol{x}_{j,1}\| + \sum_{i=1}^{n}\|\boldsymbol{x}_{i,1} - \boldsymbol{x}_{\mathrm{avg},1}\| + n^2M\Gamma_1\sum_{t=1}^{T-1}\sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\alpha_s\sigma_1^{(t-s)K_1+K_t-l-1}$$
$$+ n^2M\Gamma_1\sum_{t=1}^{T-1}\sum_{s=1}^{t}\alpha_s\sigma_1^{(t-s)K_1-1} + 2nM\sum_{t=1}^{T-1}\alpha_t. \tag{A6}$$

Now, we are going to handle the summations involved in (A6). Recalling $0 < \sigma_1 < 1$ and $K_1 \geqslant 2$, we have $\sum_{t=1}^{T-1}\sigma_1^{tK_1-1} = \frac{1}{\sigma_1}\sum_{t=1}^{T-1}(\sigma_1^{K_1})^t \leqslant \frac{1}{1-\sigma_1^{K_1}}$. Note that the sequence $\{K_t\}$ is non-decreasing. Then, it is not difficult to obtain $\sum_{t=1}^{T-1}\sum_{s=1}^{t}\sum_{l=1}^{K_s-1}\alpha_s\sigma_1^{(t-s)K_1+K_t-l-1} \leqslant \sum_{t=1}^{T-1}\sum_{s=1}^{t}\alpha_s\sigma_1^{(t-s)K_1}\sum_{l=1}^{K_t-1}\sigma_1^{K_t-l-1} \leqslant \frac{1}{(1-\sigma_1)(1-\sigma_1{}^{K_1})}\sum_{t=1}^{T}\alpha_t$. Similarly, it can be verified that $\sum_{t=1}^{T-1}\sum_{s=1}^{t}\alpha_s\sigma_1^{(t-s)K_1-1} \leqslant \frac{1}{\sigma_1(1-\sigma_1{}^{K_1})}\sum_{t=1}^{T}\alpha_t$. Combining the above conditions, (A5), and (A6) yields the condition in (11).

Appendix B Proof of Lemma 2

Under Assumption 4, we have

$$\sum_{t=1}^{T}\sum_{k=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\widehat{\nabla}f_{i,t}^{k} - \frac{1}{n}\nabla F_t(\boldsymbol{x}_{\mathrm{avg},t}^{k})\right\|$$

$$\begin{aligned}
&\leqslant \sum_{t=1}^{T}\sum_{k=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\widehat{\nabla} f_{i,t}^k-\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)\right\|+\sum_{t=1}^{T}\sum_{k=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)-\frac{1}{n}\nabla F_t(\boldsymbol{x}_{\mathrm{avg},t}^k)\right\| \\
&\leqslant \sum_{t=1}^{T}\sum_{k=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\widehat{\nabla} f_{i,t}^k-\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)\right\|+G_X\sum_{t=1}^{T}\sum_{k=1}^{K_t}\sum_{i=1}^{n}\alpha_t\left\|\boldsymbol{x}_{i,t}^k-\boldsymbol{x}_{\mathrm{avg},t}^k\right\|, \quad \text{(B1)}
\end{aligned}$$

where the last inequality is obtained by using the double stochasticity of A_t. Then, we firstly analyse the first term on the right hand side.

$$\begin{aligned}
\widehat{\nabla} f_{i,t}^k &= \sum_{j=1}^{n}[A_t]_{ij}\widehat{\nabla} f_{j,t}^{k-1}+\sum_{j=1}^{n}[A_t]_{ij}\delta_{j,t}^{k-1} \\
&= \sum_{j=1}^{n}[A_t^k]_{ij}\nabla f_{j,t}(\hat{\boldsymbol{x}}_{j,t}^1)+\sum_{l=2}^{k}\sum_{j=1}^{n}[A_t^{k-l+1}]_{ij}\delta_{j,t}^{l-1}. \quad \text{(B2)}
\end{aligned}$$

Based on Algorithm 1, it is obtained that $\sum_{i=1}^n \overline{\nabla} f_{i,1}^1=\sum_{i=1}^n \nabla f_{i,1}(\hat{\boldsymbol{x}}_{i,1}^1)$. Now we assume $\sum_{i=1}^n \overline{\nabla} f_{i,t}^{k-1}=\sum_{i=1}^n \nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^{k-1})$. Then, we intend to prove that the equality also holds at iteration $k+1$. Actually, $\sum_{i=1}^n \overline{\nabla} f_{i,t}^k=\sum_{i=1}^n \widehat{\nabla} f_{i,t}^{k-1}+\sum_{i=1}^n \nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)-\sum_{i=1}^n \nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^{k-1})=\sum_{i=1}^n\sum_{j=1}^n [A_t]_{ij}\overline{\nabla} f_{j,t}^{k-1}+\sum_{i=1}^n \nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)-\sum_{i=1}^n \overline{\nabla} f_{i,t}^{k-1}=\sum_{i=1}^n \nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)$, where the third equality combines the double stochasticity of A_t.

This further gives

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}[A_t]_{ij}\overline{\nabla} f_{j,t}^{k-1}+\frac{1}{n}\sum_{i=1}^{n}\delta_{i,t}^{k-1} \\
&= \frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^{k-1})+\frac{1}{n}\sum_{i=1}^{n}\delta_{i,t}^{k-1} \\
&= \frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^1)+\frac{1}{n}\sum_{l=2}^{k}\sum_{i=1}^{n}\delta_{i,t}^{l-1}. \quad \text{(B3)}
\end{aligned}$$

Based on the above analysis, we obtain, for any $k\geqslant 2$,

$$\begin{aligned}
\left\|\widehat{\nabla} f_{i,t}^k-\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)\right\| &\leqslant \sum_{j=1}^{n}\left|[A_t^k]_{ij}-\frac{1}{n}\right|\left\|\nabla f_{j,t}(\hat{\boldsymbol{x}}_{j,t}^1)\right\|+\sum_{l=2}^{k}\sum_{j=1}^{n}\left|[A_t^{k-l+1}]_{ij}-\frac{1}{n}\right|\left\|\delta_{j,t}^{l-1}\right\| \\
&\leqslant \Gamma_1{\sigma_1}^{k-1}\sum_{j=1}^{n}\left\|\nabla f_{j,t}(\hat{\boldsymbol{x}}_{j,t}^1)\right\|+\Gamma_1\sum_{l=2}^{k}\sum_{j=1}^{n}{\sigma_1}^{k-l}\left\|\delta_{j,t}^{l-1}\right\|. \quad \text{(B4)}
\end{aligned}$$

This implies

$$\begin{aligned}
&\sum_{k=1}^{K_t}\sum_{i=1}^{n}\left\|\widehat{\nabla} f_{i,t}^k-\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)\right\| \\
&\leqslant \sum_{i=1}^{n}\left\|\widehat{\nabla} f_{i,t}^1-\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^1)\right\|+n\Gamma_1\sum_{k=2}^{K_t}\sum_{j=1}^{n}{\sigma_1}^{k-1}\left\|\nabla f_{j,t}(\hat{\boldsymbol{x}}_{j,t}^1)\right\|+n\Gamma_1\sum_{k=2}^{K_t}\sum_{l=2}^{k}\sum_{j=1}^{n}{\sigma_1}^{k-l}\left\|\delta_{j,t}^{l-1}\right\|. \quad \text{(B5)}
\end{aligned}$$

Now we are going to estimate the bounds of the terms in (B5). Firstly, we have $\sum_{i=1}^n\|\widehat{\nabla} f_{i,t}^1-\frac{1}{n}\sum_{i=1}^n\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^1)\|\leqslant\sum_{i=1}^n\sum_{j=1}^n|[A_t]_{ij}-\frac{1}{n}|\|\nabla f_{j,t}(\hat{\boldsymbol{x}}_{j,t}^1)\|\leqslant n^2\Gamma_1 L_X$. Next, it is easily obtained that $n\Gamma_1\sum_{k=2}^{K_t}\sum_{j=1}^n{\sigma_1}^{k-1}\|\nabla f_{j,t}(\hat{\boldsymbol{x}}_{j,t}^1)\|\leqslant\frac{\sigma_1 n^2\Gamma_1 L_X}{1-\sigma_1}$. Moreover, it can be verified that

$$\begin{aligned}
n\Gamma_1\sum_{k=2}^{K_t}\sum_{l=2}^{k}\sum_{j=1}^{n}{\sigma_1}^{k-l}\left\|\delta_{j,t}^{l-1}\right\| &\leqslant n\Gamma_1\left(\sum_{k=2}^{K_t}{\sigma_1}^{k-2}\right)\left(\sum_{k=2}^{K_t}\sum_{i=1}^{n}\left\|\delta_{i,t}^{k-1}\right\|\right) \\
&\leqslant \frac{n\Gamma_1 G_X}{1-\sigma_1}\sum_{k=1}^{K_t-1}\sum_{i=1}^{n}\left\|\hat{\boldsymbol{x}}_{i,t}^{k+1}-\hat{\boldsymbol{x}}_{i,t}^k\right\|. \quad \text{(B6)}
\end{aligned}$$

Note that $\sum_{i=1}^n\|\hat{\boldsymbol{x}}_{i,t}^{k+1}-\hat{\boldsymbol{x}}_{i,t}^k\|\leqslant\sum_{i=1}^n\|\hat{\boldsymbol{x}}_{i,t}^{k+1}-\boldsymbol{x}_{\mathrm{avg},t}^k\|+\sum_{i=1}^n\|\boldsymbol{x}_{\mathrm{avg},t}^k-\hat{\boldsymbol{x}}_{i,t}^k\|\leqslant\sum_{i=1}^n\|\boldsymbol{x}_{i,t}^{k+1}-\boldsymbol{x}_{\mathrm{avg},t}^k\|+\sum_{i=1}^n\|\boldsymbol{x}_{\mathrm{avg},t}^k-\boldsymbol{x}_{i,t}^k\|\leqslant\sum_{i=1}^n\|\hat{\boldsymbol{x}}_{i,t}^k+\alpha_t(\boldsymbol{v}_{i,t}^k-\hat{\boldsymbol{x}}_{i,t}^k)-\boldsymbol{x}_{\mathrm{avg},t}^k\|+\sum_{i=1}^n\|\boldsymbol{x}_{\mathrm{avg},t}^k-\boldsymbol{x}_{i,t}^k\|\leqslant 2\sum_{i=1}^n\|\boldsymbol{x}_{i,t}^k-\boldsymbol{x}_{\mathrm{avg},t}^k\|+\alpha_t nM$. This, together with (B6), implies

$$n\Gamma_1\sum_{k=2}^{K_t}\sum_{l=2}^{k}\sum_{j=1}^{n}{\sigma_1}^{k-l}\left\|\delta_{j,t}^{l-1}\right\|\leqslant\frac{2n\Gamma_1 G_X}{1-\sigma_1}\sum_{k=1}^{K_t-1}\sum_{i=1}^{n}\left\|\boldsymbol{x}_{i,t}^k-\boldsymbol{x}_{\mathrm{avg},t}^k\right\|+\frac{n^2\Gamma_1 G_X M}{1-\sigma_1}\alpha_t K_t. \quad \text{(B7)}$$

Then, substituting the above conditions and (B7) into (B5), we have

$$\sum_{t=1}^{T}\alpha_t\sum_{k=1}^{K_t}\sum_{i=1}^{n}\left\|\widehat{\nabla} f_{i,t}^k-\frac{1}{n}\sum_{i=1}^{n}\nabla f_{i,t}(\hat{\boldsymbol{x}}_{i,t}^k)\right\|\leqslant\frac{n^2\Gamma_1 L_X}{1-\sigma_1}\sum_{t=1}^{T}\alpha_t+\frac{n^2\Gamma_1 G_X M}{1-\sigma_1}\sum_{t=1}^{T}\alpha_t^2 K_t+\frac{2n\Gamma_1 G_X}{1-\sigma_1}\sum_{t=1}^{T}\sum_{k=1}^{K_t-1}\sum_{i=1}^{n}\alpha_t\left\|\boldsymbol{x}_{i,t}^k-\boldsymbol{x}_{\mathrm{avg},t}^k\right\|. \quad \text{(B8)}$$

On the other hand, similar to Lemma 1, we easily obtain the results: (1) $\boldsymbol{x}_{i,t}^{k+1}=\sum_{j=1}^n[A_t^k]_{ij}\boldsymbol{x}_{j,t}^1+\alpha_t\sum_{l=1}^{k-1}\sum_{j=1}^n[A_t^{k-l}]_{ij}(\boldsymbol{v}_{j,t}^l-\hat{\boldsymbol{x}}_{j,t}^l)+\alpha_t(\boldsymbol{v}_{i,t}^k-\hat{\boldsymbol{x}}_{i,t}^k)$; (2) $\boldsymbol{x}_{\text{avg},t}^{k+1}=\frac{1}{n}\sum_{i=1}^n\boldsymbol{x}_{i,t}^1+\frac{\alpha_t}{n}\sum_{l=1}^{k-1}\sum_{i=1}^n(\boldsymbol{v}_{i,t}^l-\hat{\boldsymbol{x}}_{i,t}^l)+\alpha_t(\boldsymbol{v}_{\text{avg},t}^k-\boldsymbol{x}_{\text{avg},t}^k)$. Based on these two conditions, we further obtain

$$\begin{aligned}\left\|\boldsymbol{x}_{i,t}^{k+1}-\boldsymbol{x}_{\text{avg},t}^{k+1}\right\| &\leqslant \Gamma_1{\sigma_1}^{k-1}\sum_{j=1}^n\|\boldsymbol{x}_{j,t}^1\|+\alpha_t nM\Gamma_1\sum_{l=1}^{k-1}{\sigma_1}^{k-l-1}+2\alpha_tM\\ &=\Gamma_1{\sigma_1}^{k-1}\sum_{j=1}^n\|\boldsymbol{x}_{j,t}^1-\boldsymbol{x}_{j,1}+\boldsymbol{x}_{j,1}\|+\alpha_t n\Gamma_1M\sum_{s=1}^{k-1}{\sigma_1}^{s-1}+2\alpha_tM\\ &\leqslant nM\Gamma_1{\sigma_1}^{k-1}+\Gamma_1{\sigma_1}^{k-1}\sum_{j=1}^n\|\boldsymbol{x}_{j,1}\|+\alpha_t n\Gamma_1M\sum_{s=1}^{k-1}{\sigma_1}^{s-1}+2\alpha_tM.\end{aligned}\tag{B9}$$

Summing (B9) with respect to i,k, and t, we have

$$\begin{aligned}&\sum_{t=1}^T\sum_{k=1}^{K_t}\sum_{i=1}^n\alpha_t\|\boldsymbol{x}_{i,t}^k-\boldsymbol{x}_{\text{avg},t}^k\|\\ &\leqslant\sum_{t=1}^T\sum_{i=1}^n\alpha_t\|\boldsymbol{x}_{i,t}^1-\boldsymbol{x}_{\text{avg},t}^1\|+\sum_{t=1}^T\sum_{k=2}^{K_t}\sum_{i=1}^n\alpha_t\|\boldsymbol{x}_{i,t}^k-\boldsymbol{x}_{\text{avg},t}^k\|\\ &\leqslant nM\sum_{t=1}^T\alpha_t+n^2M\Gamma_1\sum_{t=1}^T\sum_{k=2}^{K_t}{\sigma_1}^{k-2}\alpha_t+n\Gamma_1\sum_{t=1}^T\sum_{k=2}^{K_t}\sum_{j=1}^n{\sigma_1}^{k-2}\alpha_t\|\boldsymbol{x}_{j,1}\|+n^2\Gamma_1M\sum_{t=1}^T\sum_{k=2}^{K_t}\sum_{s=1}^{k-1}{\sigma_1}^{s-1}\alpha_t^2+2nM\sum_{t=1}^T\alpha_t^2K_t\\ &\leqslant\left[nM+\frac{n\Gamma_1}{1-\sigma_1}\sum_{j=1}^n(\|\boldsymbol{x}_{j,1}\|+M)\right]\sum_{t=1}^T\alpha_t+\left(\frac{n^2\Gamma_1M}{1-\sigma_1}+2nM\right)\sum_{t=1}^T\alpha_t^2K_t.\end{aligned}\tag{B10}$$

Combining (B1), (B8), and (B10) yields (12).

Appendix C Proof of Lemma 3

Based on Algorithm 1 and the smooth property in Assumption 4, we have

$$\begin{aligned}F_t\left(\boldsymbol{x}_{\text{avg},t}^{k+1}\right)-F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right) &\leqslant\left\langle\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right),\boldsymbol{x}_{\text{avg},t}^{k+1}-\boldsymbol{x}_{\text{avg},t}^k\right\rangle+\frac{nG_X}{2}\left\|\boldsymbol{x}_{\text{avg},t}^{k+1}-\boldsymbol{x}_{\text{avg},t}^k\right\|^2\\ &=\alpha_t\left\langle\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right),\boldsymbol{v}_{\text{avg},t}^k-\boldsymbol{x}_{\text{avg},t}^k\right\rangle+\frac{n\alpha_t^2G_X}{2}\left\|\boldsymbol{v}_{\text{avg},t}^k-\boldsymbol{x}_{\text{avg},t}^k\right\|^2\\ &\leqslant\alpha_t\sum_{i=1}^n\left\langle\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right),\boldsymbol{v}_{i,t}^k-\boldsymbol{x}_{\text{avg},t}^k\right\rangle+\frac{n\alpha_t^2G_XM^2}{2},\end{aligned}\tag{C1}$$

where we used the fact that $\boldsymbol{x}_{\text{avg},t}^{k+1}=\boldsymbol{x}_{\text{avg},t}^k+\alpha_t(\boldsymbol{v}_{\text{avg},t}^k-\boldsymbol{x}_{\text{avg},t}^k)$. Furthermore, it is noted that

$$\begin{aligned}\left\langle\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right),\boldsymbol{v}_{i,t}^k-\boldsymbol{x}_{\text{avg},t}^k\right\rangle &\leqslant\left\langle\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)-\widehat{\nabla}f_{i,t}^k,\boldsymbol{v}_{i,t}^k-\boldsymbol{x}_{\text{avg},t}^k\right\rangle+\left\langle\widehat{\nabla}f_{i,t}^k,\boldsymbol{x}_t^*-\boldsymbol{x}_{\text{avg},t}^k\right\rangle\\ &\leqslant 2M\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)-\widehat{\nabla}f_{i,t}^k\right\|+\left\langle\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right),\boldsymbol{x}_t^*-\boldsymbol{x}_{\text{avg},t}^k\right\rangle\\ &\leqslant 2M\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)-\widehat{\nabla}f_{i,t}^k\right\|+\frac{1}{n}\left[F_t\left(\boldsymbol{x}_t^*\right)-F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)\right],\end{aligned}\tag{C2}$$

where the first inequality and the last inequality utilize the optimality condition of the variable $\boldsymbol{v}_{i,t}^k$ and the convexity condition of $F_t(\boldsymbol{x})$, respectively. Subtracting the term $F_t(x_t^*)$ from both sides of (C1) and substituting (C2) into (C1), the following inequality is obtained:

$$\begin{aligned}&F_t\left(\boldsymbol{x}_{\text{avg},t}^{k+1}\right)-F_t(x_t^*)\\ &=F_t\left(\boldsymbol{x}_{\text{avg},t}^{k+1}\right)-F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)+F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)-F_t(x_t^*)\\ &\leqslant(1-\alpha_t)\left[F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)-F_t(x_t^*)\right]+2\alpha_tM\sum_{i=1}^n\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^k\right)-\widehat{\nabla}f_{i,t}^k\right\|+\frac{n\alpha_t^2G_XM^2}{2}\\ &=(1-\alpha_t)^k\left[F_t\left(\boldsymbol{x}_{\text{avg},t}^1\right)-F_t(x_t^*)\right]+2\alpha_tM\sum_{l=0}^{k-1}(1-\alpha_t)^l\sum_{i=1}^n\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^{k-l}\right)-\widehat{\nabla}f_{i,t}^{k-l}\right\|+\frac{n\alpha_t^2G_XM^2}{2}\sum_{l=0}^{k-1}(1-\alpha_t)^l\\ &\leqslant(1-\alpha_t)^k\left[F_t\left(\boldsymbol{x}_{\text{avg},t}^1\right)-F_t(x_t^*)\right]+2\alpha_tM\sum_{l=1}^{k}\sum_{i=1}^n\left\|\frac{1}{n}\nabla F_t\left(\boldsymbol{x}_{\text{avg},t}^l\right)-\widehat{\nabla}f_{i,t}^l\right\|+\frac{n\alpha_tG_XM^2}{2},\end{aligned}\tag{C3}$$

where in the last inequality we utilize the two facts $(1-\alpha_t)^l\leqslant 1$ and $\sum_{l=0}^{k-1}(1-\alpha_t)^l\leqslant\frac{1}{\alpha_t}$.