Randomised block-coordinate Frank-Wolfe algorithm for distributed online learning over networks

Jingchao Li, Qingtao Wu, Ruijuan Zheng, Junlong Zhu, Quanbo Ge, Mingchuan Zhang

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, People’s Republic of China
School of Electronic and Information Engineering, Tongji University, Shanghai 200092, People’s Republic of China
E-mail: zhang_mch@haust.edu.cn

Abstract: The distributed online algorithms which are based on the Frank-Wolfe method can effectively deal with constrained optimisation problems. However, the calculation of the full (sub)gradient vector in those algorithms leads to a huge computational cost at each iteration. To reduce the computational cost of the algorithms mentioned above, the authors present a distributed online randomised block-coordinate Frank-Wolfe algorithm over networks. Each agent in the networks only needs to calculate a subset of the coordinates of its (sub)gradient vector in this algorithm. Furthermore, they make a detailed theoretical analysis of the regret bound of this algorithm. When all local objective functions satisfy the conditions of strongly convex functions, the authors’ algorithm attains the regret bound of $O(T^{1/4})$, where $T$ is the total number of iterations. Furthermore, the theorem results are verified via simulation experiments, which show that the algorithm is effective and efficient.

1 Introduction

Distributed optimisation has developed unprecedentedly over the past years. Many practical problems can be solved by modelling them to distributed optimisation problems. For example, the problems of large-scale machine learning [1–3], estimation and detection in sensor networks [4, 5], resource allocation in networks [6, 7], the control of intelligent power grid [8, 9] and the problem of multi-agent coordination and consensus in networks [10, 11]. To solve the problems mentioned above and other related problems, some optimisation algorithms are pre-requisite. However, how to design an efficient optimisation algorithm is vitally significant. Meanwhile, the computing power of a single agent in the networks is limited. Therefore, we need to design distributed optimisation algorithms that take advantage of the collaboration of multiple agents to solve large-scale optimisation problems. Many distributed optimisation algorithms have been designed and applied to practical application problems [12–15].

Although these algorithms mentioned above have achieved great success in dealing with distributed optimisation problems. However, these methods consider the case that the objective function of each agent is fixed with time. Nevertheless, the actual network environment is dynamic, which leads to the objective function of each agent that may change with time. Therefore, it is very meaningful to research the dynamic objective function in distributed optimisation problems. To address distributed online optimisation problems, distributed online optimisation algorithms also have received a large amount of research and achieved great success in recent years [16–21].

In practice, distributed optimisation problems are constrained generally. To solve the constrained problems, some projection operations are essential. Moreover, the number of dimensions of data is very large. When tackling these data, the computational cost of the projection operation is expensive. For this reason, the Frank-Wolfe algorithm in [22] is well known for addressing large-scale constrained optimisation problems. In this algorithm, the expensive projection operation steps are replaced by more efficient linear optimisation steps. Zhang et al. [23] extended the Frank-Wolfe algorithm to a distributed online learning setting and proposed a distributed online projection-free algorithm. In addition, the regret function is used to quantise the performance of an online algorithm and it depends on time $T$ [24]. The regret of the algorithm in [23] is $O(T^{1/2})$. To improve the regret bound, a novel distributed online Frank-Wolfe algorithm was proposed in [25], which reaches a regret bound of $O(T^{1/4})$.

As mentioned previously, these distributed online algorithms based on the Frank-Wolfe method can reduce the amount of computation of each agent at each iteration. However, each agent needs to calculate the full (sub)gradient vectors, which may incur the costly computation when dealing with high-dimensional data sets. To reduce the computational cost, some block-coordinate algorithms were presented in [26–28]. In this paper, we utilise the randomised coordinate descent algorithm and Frank-Wolfe method to further reduce the computational cost. A distributed online randomised block-coordinate Frank-Wolfe (DORBCFW) algorithm is proposed to solve the large scale constrained optimisation problems. In this algorithm, each agent only randomly selects a subset of the (sub) gradient vector and updates the decision along with the (sub) gradient direction in the subset at each iteration; the projection operation is replaced by a linear optimisation step.

The main contributions of this paper are concluded as follows:

- We present a randomised block-coordinate Frank-Wolfe algorithm to address a distributed online constraint optimisation problem. This algorithm uses the advantage of the randomised block-coordinate method to reduce the calculation of full (sub)gradient in each iteration process.
- We analyse the convergence of the algorithm in dealing with distributed strongly convex optimisation problem by precise mathematical derivation. When the object function of each agent satisfies the condition of a strongly convex function, we prove that the proposed algorithm reaches the regret $O(T^{1/4})$.
- We also verify the theorem results via simulation experiments. The results show that the proposed algorithm is effective and efficient.

The rest of this paper is organised as follows. We explain the meanings of some mathematical symbols in this paper and introduce related background in Section 2. We describe the research problem and present our method in Section 3. In Section 4, we put forward the presupposition of proof and give the main results of this paper. The performance of the algorithm is proved by detailed mathematical analysis in Section 5. The simulation experiments are provided in Section 6. We give a conclusion of this paper in Section 7.
1. Input: convex set ℳ ; maximum iteration number T; doubly stochastic matrix B = [b_{ij}] ∈ ℝ^{n×n}; diagonal matrix W(t) ∈ ℝ^{d×d}.
2. Initialize: g_0(t) = W(t)∇f_t(i)z(t), W(t) = I_d, ∀t ∈ V.
3. Output: x_i(t), t ∈ {1, 2, ..., T}, ∀t ∈ V.
4. for t = 1, 2, ..., T do
  5. for each agent i ∈ V do
  6. Consensus step:
  7. z_i(t) = ∑{j ∈ N(i)} b_{ij}x_j(t).
  8. Aggregating step:
  9. g_{t,i} = ∑{j ∈ N(i)} b_{ij}g_{t,i}(t−1) + W(t)∇f_t(i)z_i(t) − W(t−1)∇f_t(i)z_i(t−1).
  10. G_{t,i}(t) = 1/n ∑_{t=1}^{T} g_{t,i}(t−1).
  11. Frank-Wolfe step:
  12. u_i(t) = arg min_{u ∈ ℳ}(G_{t,i}(t), u).
  13. x_i(t + 1) = z_i(t) + γ_t(u_i(t) − z_i(t)).
  14. end for
15. end for

Fig. 1 Algorithm 1: DORBCFW algorithm

2 Preliminaries

In this section, for easy understanding, we explain a few common notations in the paper and introduce some associated mathematical preliminaries.

2.1 Notations

Without a special declaration, all vectors are represented by column vectors in this paper. The set of a real number is indicated by ℝ. ℝ^d represents the d-dimensional real vectors or the Euclidean space. The diameter of the constraint set ℳ is expressed by sup_{z ∈ ℳ}||z||. All vectors in the Euclidean space are represented in bold. All scalars are represented in regular fonts. For example, x is a vector and y is a scalar. The inner product between vectors is represented by (·, ·). The set of positive real number is indicated by ℝ^+. Positive integers set is indicated by ℤ^+$. The expectations of random variables are expressed as E[·]. The column vector of all ones is represented by 1. Notation I denotes the identity matrix. ℳ is a bounded and convex set in the Euclidean space.

2.2 Graph

We research a network, which is composed by nodes represented by ℐ. ℐ^d represents the d-dimensional real vectors or the Euclidean space. Each agent in the network corresponds to a vertex in the graph. The vertex set is denoted by V. Two vertices on edge can send messages to each other. The set of nodes is denoted by N(i), i.e. (i, j) ∈ E. All agents directly connect to the agent i are called neighbours of an agent i; the neighbour set of the agent i is represented by N(i). The communication relationship between agents is represented by a doubly stochastic matrix B = [b_{ij}^n]^{n×n}, which satisfies ∑_{j ∈ V} b_{ij} = 1 and ∑_{i ∈ V} b_{ij} = 1 for j ∈ V and i ∈ V.

2.3 Regret

We usually use regret to measure an online algorithm. Regret is the cumulative error between the loss of the decision chosen in each iteration and the loss of the current best decision. It is defined in [25] as follows:

\[ R_T(\bar{x}(t), x^*) = \frac{1}{n} \sum_{i = 1}^{n} \sum_{t = 1}^{T} f_i(x(t)) - \frac{1}{n} \sum_{i = 1}^{n} \sum_{t = 1}^{T} f_i(x^*), \]

where \( \bar{x}(t) = (1/n) \sum_{i = 1}^{n} x_i(t) \) is a decision with an average meaning for all nodes at time t and \( x^* \) is the best decision after the fact. The regret is a sublinear function about T for a well performance online algorithm.

2.4 Definitions

We propose some definitions of function. In these definitions, ℳ denotes a convex and bounded set. ℳ is also a subset of ℝ^d, i.e. ℳ ⊆ ℝ^d.

**Definition 1:** For all \( x, z ∈ ℳ \), if a function \( f \) satisfies all the property of convex function and \( α ∈ [0, 1] \), we obtain

\[ f(αx + (1-α)z) \leq αf(x) + (1-α)f(z). \]  

**Definition 2:** If the function \( f \) is termed as \( β \)-smooth, when \( x, z ∈ ℳ \) and \( β > 0 \), satisfy

\[ f(z) ≤ f(x) + ⟨∇f(x), z − x⟩ + \frac{1}{2β} ||z − x||^2. \]

From Definition 2, we can attain the following equivalence inequality:

\[ ||∇f(x) − ∇f(z)|| ≤ β ||x − z||. \]

**Definition 3:** When \( x, z ∈ ℳ \) and \( μ ≥ 0 \), if a function \( f \) is μ-strongly convex, then satisfies

\[ f(z) ≥ f(x) + ⟨∇f(x), z − x⟩ + \frac{1}{2μ} ||z − x||^2. \]

Let \( x^* = arg \min_{z ∈ ℳ} f(x) \), from Definition 3; we can attain

\[ f(x) − f(x^*) ≥ μ ||x − x^*||^2. \]

**Definition 4:** For any \( x, z ∈ ℳ \) and \( L ∈ ℜ^+ \), if \( f \) is \( L \)-Lipschitz function, then satisfies

\[ |f(x) − f(z)| ≤ L ||x − z||. \]

3 Problem description and algorithm design

In this section, we describe the problem, which is a distributed online learning problem. To address this problem, we present a distributed online algorithm.

The local functions of all agents change over time in this problem. At iteration t, each agent i in the networks chooses a strategy \( x_i(t) \) from ℳ, where ℳ ⊆ ℜ^d is a compact and convex set. Each agent i suffers a loss function \( f_i : ℳ → ℜ \), which is generated by the adversary after it makes a decision. The loss suffered by the agent i is defined as \( f_i(x_i(t)) \). The goal is to minimise a global loss function through interaction and cooperation between agents in the network. The problem is formulated as follows:

\[ \min_{x ∈ ℳ} f(x) = \frac{1}{n} \sum_{i = 1}^{n} \sum_{t = 1}^{T} f_i(x_i). \]

Each agent has the corresponding function information and can exchange this information with its neighbours. To address this problem, we design a DORBCFW algorithm, which is summarised in Algorithm 1 (see Fig. 1).

We divide Algorithm 1 (Fig. 1) into three steps. First of all, each agent i performs the following iterations:

\[ z_i(t) = ∑_{j ∈ N(i)} b_{ij}x_j(t), \]

where \( b_{ij} \) is a weight. Next, each agent i performs the following iterations:
where $W(t) \in \mathbb{R}^{d \times d}$ is a diagonal matrix and each diagonal element of the matrix is a Bernoulli random variable, which is represented by $w(k)$. When the Bernoulli random variable is equal to 1 with probability $p_j$, i.e., $\text{Prob}(w(k) = 1) = p_j$, for $i \in V$, $t = 1, \ldots, T$ and $k = 1, \ldots, d$. In addition, let $0 < p_j \leq 1$ for each agent $i$ in the network. The form of this diagonal matrix can be represented by

$$W(t) = \text{diag}(w_1(1), w_1(2), \ldots, w_1(d)).$$

(12)

In the end, each agent $i$ updates its decision as follows:

$$u_i(t) = \arg \min_{\bar{u} \in \mathcal{U}} \left( G_i(f_i(t), \bar{u}) \right),$$

(13)

$$x_i(t+1) = z_i(t) + \gamma_i(u_i(t) - z_i(t)),$$

(14)

where $\gamma_i \in (0, 1]$ denotes a step size. In addition, we set the initial conditions $g_i(0) = W(0)V f_i(z_i(0))$ and $W_i(0) = I_d$. Moreover, we define the regret as follows:

$$R_T(x, x) = \frac{1}{n} \sum_{t=1}^{T} \sum_{i=1}^{n} E[f_i(x(t))]_{\mathcal{H}_{t-1}} - \min_{x \in \mathcal{U}} f(x),$$

(15)

where $\mathcal{H}_{t-1}$ is the history information of the random variables until time $t-1$.

4 Assumptions and results

In this section, in order to analyse the performance of our algorithm, we present some assumptions. Furthermore, we propose the main results in this paper. To ensure communication between agents, we cite Assumption 1 in [25].

Assumption 1: We assume the communication between agents as a doubly random matrix $B = [b_{ij}]_{i \times j}$, each term of the doubly stochastic matrix $B$ satisfies $b_{ij} > 0$ and $b_{ij} > 0$ only if $i, j \in V$ and $(i, j) \in E$.

We denote the spectral radius of a matrix by $\sigma(\cdot)$. From Assumption 1, we have $\sigma(B - (1/n)11') < 1$. Hence, $\exists \lambda_0 \in (0, 1)$, for $x \in \mathbb{R}^n$ and $\bar{x} = (1/n)1'x$, we have

$$\| Bx - \bar{x} \|_2 \leq \frac{\lambda_0}{\| x - \bar{x} \|_2}.$$

(16)

We set two parameters $t_0$ and $\epsilon$, where $\exists \xi_0 \in \mathbb{R}^n$ and $\epsilon \in (0, 1]$. Furthermore, these parameters satisfy the following relation:

$$\lambda_0 \leq \frac{\xi_0}{t_0} + \frac{\xi_0}{(\xi_0 + 1)} \epsilon.$$

(17)

From (17), we can obtain

$$\Psi(t) \leq \min \left\{ 9 \left( 2C_1 \| D \| + 2D(U_2 + 2\beta \sigma \| U_1 \|) \right), \right.$$  

$$\left. \left( 2\epsilon \| \epsilon \epsilon \| - 4U(9/8) + \sqrt{4\epsilon \| \epsilon \epsilon \|^2} - (16/9)(\epsilon \| \epsilon \epsilon \|)U \right) 1 \right\}$$

(20)

Each agent uses the three steps of the proposed algorithm to generate corresponding random variables in each iteration. Moreover, we assume that the relation between random variables is as follows.

Assumption 2: The random variables are independent of each other for all nodes and blocks at each iteration, i.e. $w_j(d)$ and $w_j(k)$ are mutually independent. The random variables are independent between different iteration times, i.e. $\{w_j(d)\}$ are independent of $\mathcal{F}_{t-1}$.

Assumption 3: The set $\mathcal{U}$ is bounded and convex. Furthermore, we represent the optimal set denoted by $\mathcal{U}^*$, which is non-empty. $x^*$ is an interior point of $\mathcal{U}$. We assume that $x^*(t) \in int(\mathcal{U}, int(\mathcal{U}))$ is the set of all points in the constraint set $\mathcal{U}$. We use $\partial \mathcal{U}$ to denote the boundary set of $\mathcal{U}$. We also set $\eta = \inf_{x \in \partial \mathcal{U}} \| x - x^* \| > 0$.

Assumption 4: A function $f_{ij}$ denotes the loss function of each agent in the network and satisfies the condition of $\beta$-smooth function and $L$-Lipschitz function at any time.

Next, we set

$$\Psi(t) = E[F_i(x(t))]_{\mathcal{H}_{t-1}} - F_i(x^*(t)).$$

(19)

where $x^*(t) = \arg \min_{\bar{x} \in \mathcal{U}} F_i(\bar{x})$,

$$F_i = (1/t) \sum_{t=1}^{T} f_i,$$

and $f_i = (1/n) \sum_{x=1}^{n} f_{ij}$. Therefore, $\Psi(t)$ is upper bounded, which is presented as Theorem 1.

Theorem 1: Suppose that Assumptions 1–4 hold. Meanwhile, $f_{ij}$ is $\mu$-strong function and $\mu > 0$. We define a parameter $t^*$ in (73). When $\exists \theta_0 \in X^*$, step-size $\gamma_t = 2/(t^2 + 2)$ and $t \geq t^*$, we obtain (see (20)) , where $p_{max} = \min_{x \in \mathcal{U}} \| v \|_2 < \infty$, $p_{max} = \max_{x \in \mathcal{U}} \| v \|_2$ and $\epsilon > 1$.

$$U_1 = \sqrt{n}U_{\epsilon},$$

$$U_2 = n(2D + 2D(U_2 + 2L\beta \sigma \| U_1 \|) + 2(1/e) + 2\beta \sigma \| U_1 \|)U_{\epsilon}$$

Next, the regret bound is obtained by using Theorem 1. The result is presented as Theorem 2.

Theorem 2: Suppose that Assumptions 1–3 hold. Moreover, at each iteration, the loss function of each agent in the network is denoted by $f_{ij}$, and it satisfies the condition of a convex function and $L$-Lipschitz function at any time. In addition, the loss function might be a non-smooth function. For all $x \in X^*$, we attain

$$R_T(x, x) \leq 6\sqrt{QLD_1} + \bar{F} + 1,$$

(21)

where

$$Q = \max \{ C_1, C_2 \},$$

$$C_2 = \frac{2\beta^2 L}{D} - \frac{2U''}{D} + \frac{2\epsilon U''}{D} + \frac{16}{9}(\epsilon U'')\frac{81}{8},$$

$$U_1 = n(2D + 2U_1)/(1-e) + 2L\beta \sigma \| U_1 \|, \epsilon > 1.$$

(22)

(23)

(24)

(25)
\[ U'' = \frac{2LD + 4Lp_{\text{max}}U_1 + 4DU'}{p_{\text{max}}}. \] (25)

Furthermore, we provide detailed proofs of the two theorems in Section 5.

5 Performance analysis

In this section, Theorems 1 and 2 are proved in detail. Before analysing the performance of our algorithm, we present some average vectors as follows:

\[ \bar{u}(t) = \frac{1}{n} \sum_{i=1}^{n} u_i(t), \] (26)

\[ \bar{x}(t) = \frac{1}{n} \sum_{i=1}^{n} x_i(t), \] (27)

\[ r_i(t) = \frac{1}{n} \sum_{i=1}^{n} W_i(t) \nabla f_i(x_i(t)). \] (28)

According to the above equalities, we obtain Lemma 1.

Lemma 1: At each iterative, we obtain the following relations:

(i) \( \bar{g}_i(t+1) = r_{\bar{z}_i}(t+1); \)

(ii) \( \bar{x}(t+1) = (1-\gamma_t)\bar{x}(t) + \gamma_t \bar{u}(t). \)

The proof of lemma can be found in [25].

Next, we provide Lemma 2, which is crucial in our performance analysis.

Lemma 2: Suppose that Assumption 1 holds. Let \( \gamma_t = 1/t^\nu \) with \( 0 < \epsilon \leq 1 \). For \( t \geq t_0 \) and \( i \in V \), we obtain

\[ \max_{i \in V} \| z_i(t) - \bar{x}(t) \| \leq \frac{U_i}{\tau}. \] (29)

where \( U_i = \sqrt{\text{nd}f_{\text{min}}} \).

The proof of Lemma 2 can also be found in [25].

Next, we also have a crucial result, which is given by Lemma 3.

Lemma 3: Suppose that Assumption 1 holds. Let step-size \( \gamma_t = 1/t^\nu \) with \( 0 < \epsilon \leq 1 \). If \( t > t_0 \) and \( i \in V \), we attain

\[ \max_{i \in V} E[ \| G_i(t) - r_i(t) \| \bigg| \mathcal{F}_{t-1} ] \leq \frac{U_i}{\tau}. \] (30)

where \( U_i = \sqrt{\text{nd}f_{\text{min}}} \).

\( \xi \) is a positive constant and satisfies \( \xi \geq \| G_{\bar{z}_i}(1) - \bar{r}_{\bar{z}_i}(1) \|_2. \)

Proof: According to the property of the norm, the following inequality is obtained:

\[ \max_{i \in V} E[ \| G_i(t) - r_i(t) \| \bigg| \mathcal{F}_{t-1} ] \leq \sqrt{\frac{\lambda}{\epsilon} \sum_{i=1}^{n} E[ \| G_i(t) - r_i(t) \|_\infty \bigg| \mathcal{F}_{t-1} ]}. \] (31)

Hence, the proof of Lemma 3 is equivalent to proving the following relation:

\[ \sqrt{\frac{\lambda}{\epsilon} \sum_{i=1}^{n} E[ \| G_i(t) - r_i(t) \|_\infty \bigg| \mathcal{F}_{t-1} ]} \leq \frac{U_i}{\tau}. \] (32)

We also prove (24) by mathematical induction. When \( t = 1 \) and \( t = t_0 \), (24) holds. In the case of \( t \geq t_0 \), we assume that (24) also holds. Before proving (24), we introduce an auxiliary variable \( \rho f_{\bar{z}_i}(t+1) \), i.e.

\[ \rho f_{\bar{z}_i}(t+1) = W_i(t+1)\nabla f_i(x_i(t+1)) - W_i(t)\nabla f_i(x_i(t)). \] (33)

Substituting (25) into (5), the following equality is obtained:

\[ r_{\bar{z}_i}(t+1) = \sum_{j \in V(i)} h_i r_j(t) + \rho f_{\bar{z}_i}(t+1). \] (34)

Following from (6), we attain

\[ \sum_{i=1}^{n} \| G_{\bar{z}_i},(t+1) - r_{\bar{z}_i}(t+1) \|_\infty \leq \lambda t \sum_{i=1}^{n} \sum_{j \in V(i)} \rho f_{\bar{z}_i}(t+1) + \frac{1}{1+\epsilon} \xi + \frac{1}{t+1} \xi \] (35)

Moreover, using (20), we have

\[ r_{\bar{z}_i}(t+1) - \frac{t}{t+1} r_i(t) = \frac{1}{n(1+\epsilon)} \sum_{j \in V(i)} \sum_{j=1}^{n} \rho f_{\bar{z}_i}(t+1) \] (36)

where

\[ \xi_{\bar{z}_i}(t+1) = \frac{1}{n} \sum_{j \in V} W_i(t+1)\nabla f_{\bar{z}_i}(\bar{x}_i(t+1)). \] (37)

From (5) and using the triangle inequality, we obtain (see (38)).

Using Lemma 1, we also have

\[ E[ \| \rho f_{\bar{z}_i}(t+1) \|_\infty \bigg| \mathcal{F}_{t-1} ] \leq \lambda \rho_{\text{max}}(D + 2U_2) \bar{r}. \] (39)

Therefore, following (39) implies that

\[ E[ \| s_{\bar{z}_i},(t+1) - \xi_{\bar{z}_i}(t+1) \|_\infty \bigg| \mathcal{F}_{t-1} ] \leq \lambda t \| s_{\bar{z}_i},(1) - \xi_{\bar{z}_i}(1) \|_\infty + \lambda \rho_{\text{max}}(D + 2U_2) \frac{1}{t-\epsilon}. \] (40)

In addition, using (25) yields that

\[ \| G_{\bar{z}_i},(t+1) - \xi_{\bar{z}_i}(t+1) \|_\infty \leq \lambda t \| G_{\bar{z}_i},(1) - \xi_{\bar{z}_i}(1) \|_\infty + \| \rho f_{\bar{z}_i}(t+1) \|_\infty. \] (38)
\[
\begin{align*}
E \left[ \frac{1}{t+1} \left( \sum_{i=1}^{n} q_{t,i}(t+1) + \frac{1}{t} \sum_{i=1}^{n} q_{t,i}(t+1) \right) \right] \\
\leq \frac{2}{t+1} \sum_{i=1}^{n} \left( \frac{1}{n} \| D + 2U_i \|_{\beta_{p_{\infty}}} \right) + \frac{2(1+2U_i)}{t} + 1 \\
\leq 2(1+2U_i)\beta_{p_{\infty}} \frac{1}{t} + \frac{1}{t} + 1. 
\end{align*}
\]

Let
\[
\| s_{t,i}(1) - \xi_{t,i}(1) \|_{\infty} \leq \xi
\]
with \( \xi \in \mathbb{R}^* \) for any \( i \in V \). Using (30)-(38) and (41), we attain
\[
\sqrt{d} \sum_{i=1}^{n} E[\| G_{t,i}(t) - r_{t,i}(t+1) \|_{\infty}] \\
\leq \sqrt{d} \| b_{t} \|_{r_{t+1}}^2 \\
+ \frac{\beta_{p_{\infty}}(D + 2U_i)}{1 - e} \xi^2 \\
+ \sqrt{d} \| \beta_{p_{\infty}} \|_{r_{t+1}}^2 \\
\leq \frac{U_2}{t+1}. 
\]

Following the above statement, Lemma 3 is proved. \( \square \)

Next, we prove Theorem 1 by using Lemmas 1–3.

**Proof of Theorem 1:** For the convenience of expression in the following proof, we first define the constant \( U \) as follows:
\[
U = \max \left\{ \frac{1}{2} (\beta D^2 + 2D(U_i + \beta_{p_{\infty}} U_i)) \right\} \\
\left( \frac{2\eta \mu \epsilon^2}{9U} \right)^2 - (4U'/9) + \left( \frac{2\eta \mu \epsilon^2}{8/9} \right)^2 \right). 
\]

where \( U' = (2\beta D^2 + 4D(\beta_{p_{\infty}} U_i + U_i))/\beta_{p_{\infty}} \) and \( c > 1 \). Hence, the proof of Theorem 1 is equivalent to proving the inequality \( \Psi(t) \leq U(t+1) \) holds. Because the function \( F_i \) is \( \beta \)-smooth function according to Assumption 4, using the second part of Lemma 1, we have
\[
F_i(x(t+1)) - F_i(x(t)) + \langle \nabla F_i(x(t)), x(t+1) - x(t) \rangle \\
+ \frac{\beta}{2} \| x(t+1) - x(t) \|_2^2 \\
\leq F_i(x(t)) + \frac{\beta}{2} D^2 \\
+ \frac{\gamma_i}{n} \sum_{i=1}^{n} \langle \nabla F_i(x(t)), u_i(t) - x(t) \rangle. 
\]

In addition, if \( v \in \mathcal{M} \) and \( i \in V \), the following relation is attained:
\[
\langle \nabla F_i(x(t)), u_i(t) - x(t) \rangle \\
\leq \langle \nabla F_i(x(t)), u_i(t) - \bar{x}(t) \rangle \\
+ 2D G_i - \frac{1}{n} \sum_{i=1}^{n} W_i(t) \nabla F_i(x(t)) \rangle. 
\]

Therefore, the following relation is obtained:
\[
\langle \nabla F_i(x(t)), u_i(t) - \xi(t) \rangle \leq \frac{\beta_{p_{\infty}}}{p_{\infty} t} \langle \nabla F_i(x(t)), u_i(t) - \bar{x}(t) \rangle \\
+ 2D G_i - \frac{1}{n} \sum_{i=1}^{n} W_i(t) \nabla F_i(x(t)) \rangle. 
\]

where
\[
p_{\infty} = \begin{cases} 
\frac{\beta_{p_{\infty}}}{p_{\infty} t} & \text{if } \langle \nabla F_i(x(t)), u_i(t) - \bar{x}(t) \rangle \geq 0; \\
\text{otherwise.} & 
\end{cases} 
\]

Therefore, combining (47) and (49), we attain
\[
E[\xi(t+1) | \mathcal{F}_t] \leq F_i(\xi(t)) + \frac{\beta}{2} D^2 \\
+ \frac{\gamma_{\infty}}{p_{\infty} t} \langle \nabla F_i(x(t)), u_i(t) - \bar{x}(t) \rangle \\
+ \gamma_{\infty} \frac{2D U_i}{p_{\infty} t} + \gamma_{\infty} \frac{2D \beta_{p_{\infty}} U_i}{p_{\infty} t.}
\]

where \( u = \bar{u}(t) \in \arg \min_{u \in \mathcal{U}} \langle \nabla F_i(x(t)), u \rangle \). Further, from (51), we obtain
\[
E[\xi(t+1) - F_i(\xi(t)) | \mathcal{F}_t] \\
\leq F_i(x(t)) - F_i(\xi(t)) + \frac{\beta}{2} D^2 \\
- \frac{\gamma_{\infty}}{p_{\infty} t} \langle \nabla F_i(x(t)), \bar{x}(t) - \bar{u}(t) \rangle \\
+ \gamma_{\infty} \frac{2D U_i}{p_{\infty} t} + \gamma_{\infty} \frac{2D \beta_{p_{\infty}} U_i}{p_{\infty} t.}
\]

Using the definition of \( \Psi(t) \), implies that
\[
\Psi(t+1) \\
\leq \frac{t+1}{t+1} \Psi(t) - \frac{\gamma_{\infty}}{p_{\infty} t} \langle \nabla F_i(x(t)), \bar{x}(t) - \bar{u}(t) \rangle \\
+ \frac{t+1}{t+1} \left( \frac{\beta}{2} D^2 + \frac{2D U_i}{p_{\infty} t} + \gamma_{\infty} \frac{2D \beta_{p_{\infty}} U_i}{p_{\infty} t.} \right) \\
+ \frac{1}{t+1} (f_{\infty}(x(t+1)) - f_{\infty}(x(t+1))).
\]

Moreover, according to Lemma 6 in [29], we attain
\[
\langle \nabla F_i(x(t)), \bar{x}(t) - \bar{u}(t) \rangle \geq \sqrt{2\mu \rho_{\Psi}}. 
\]

Similar to [25], we know that
\[
(f_{\infty}(x(t+1)) - f_{\infty}(x(t+1))) \leq \frac{2U}{9} (t+1)^{-1/2}.
\]

Substituting (54) and (55) into (53), we attain
\[ \Psi(t+1) \leq \Psi(t) - \gamma_t \frac{\mu}{\mu_\max} \sqrt{2\mu_t \Psi(t)} + \frac{2U}{9(t+1)^{\zeta_\gamma}} \frac{1}{\beta D^2} + 2D \frac{\beta \mu_{\max} U_1 + U_2}{\mu_{\max}(t+1)^{\zeta_\gamma}}. \] (56)

When \( \sqrt{\Psi(t)} - \gamma_t \frac{\mu}{\mu_\max} \sqrt{2\mu_t \Psi(t)} \leq 0 \), we obtain
\[ \Psi(t+1) \leq \frac{U}{t+2}. \] (59)

When \( \sqrt{\Psi(t)} - \gamma_t \frac{\mu}{\mu_\max} \sqrt{2\mu_t \Psi(t)} > 0 \), according to (56) and using the following inequality:
\[ \frac{1}{t+1} - \frac{1}{t+2} \leq \frac{1}{(t+1)^2}, \] (60)
we obtain
\[ \Psi(t+1) - \frac{U}{t+1} \leq \frac{U}{t+2} - \frac{U}{t+1} - \frac{\gamma_t \sqrt{2\mu_t \Psi(t)}}{(t+1)^{\zeta_\gamma}} + \frac{2U}{9(t+1)^{\zeta_\gamma}} + \frac{2\beta D^2 + 4D \frac{\beta \mu_{\max} U_1 + U_2}{\mu_{\max}(t+1)^{\zeta_\gamma}}}{(t+1)^{\zeta_\gamma}} \leq \frac{U}{t+2} - \frac{\gamma_t \sqrt{2\mu_t \Psi(t)}}{(t+1)^{\zeta_\gamma}} + \frac{2\beta D^2 + 4D \frac{\beta \mu_{\max} U_1 + U_2}{\mu_{\max}(t+1)^{\zeta_\gamma}}}{(t+1)^{\zeta_\gamma}} \leq \frac{1}{(t+1)^{\zeta_\gamma}} \left[ \frac{U}{\sqrt{t+1}} - \frac{2U}{9} + \frac{2\beta D^2 + 4D \frac{\beta \mu_{\max} U_1 + U_2}{\mu_{\max}(1+1)^{\zeta_\gamma}}}{(1+1)^{\zeta_\gamma}} \right] \leq \frac{1}{(t+1)^{\zeta_\gamma}} \left[ \frac{U}{\sqrt{t+1}} + \frac{2U}{9}(1-\epsilon) + \frac{(2\beta D^2 + 4D \frac{\beta \mu_{\max} U_1 + U_2}{\mu_{\max}(1+1)^{\zeta_\gamma}})(1-\epsilon)}{(1+1)^{\zeta_\gamma}} \right]. \] (61)

Because \( \sqrt{t+1} \) is a monotonic decreasing function with \( t \), we define a quantity as follows:
\[ t' = \inf \left\{ t \geq 1 : \frac{U}{\sqrt{t+1}} + \frac{2U}{9}(1-\epsilon) + \frac{2\beta D^2 + 4D \frac{\beta \mu_{\max} U_1 + U_2}{\mu_{\max}(1+1)^{\zeta_\gamma}})(1-\epsilon) \leq 0 \right\}. \] (62)

the quantity \( t' \) can exist, which is due to \( \epsilon > 1 \). Therefore, if \( t > t' \), we obtain
\[ \Psi(t+1) \leq \frac{U}{t+2}. \] (63)

Thus, Theorem 1 is obtained. □

Next, the proof of Theorem 2 is as follows.

Proof of Theorem 2: In order to establish the regret bound of Algorithm 1 (Fig. 1), we define an auxiliary function as follows:
\[ f_j(x) = \langle \nabla f_j(x(t)), x \rangle + \frac{L}{D} \| x - x(t) \|_2^2, \] (64)

where the (sub)gradient of function \( f_j \) at the decision point \( x(t) \) is denoted by \( \nabla f_j(x(t)) \). According to the property of \( L \)-Lipschitz function that \( \| x - x(t) \|_2 \leq D \), we obtain
\[ \nabla f_j(x) = \nabla f_j(x(t)) + \frac{2L}{D}(x - x(t)). \] (65)

Following from the property of \( L \)-Lipschitz function and the bound of \( \| x - x(t) \|_2 \leq D \), we know that \( 3L \geq \| \nabla f_j(x) \|_2 \). Hence, we conclude that \( f_j \) is a \( 3L \)-Lipschitz function. In addition, we obtain
\[ f_j(x + y) - f_j(x) = \langle \nabla f_j(x(t)), y \rangle + \frac{L}{D} \| y \|_2^2. \] (66)

From the above equality, we know that \( f_{j_1} \) is not only \( L/D \)-smooth function but also \( L/D \)-strong convex function. Let \( x^*(t) = \arg \min_{x \in \mathcal{X}} f_j(x) \) and \( \bar{F}_j(x) = (1/n) \sum_{i=1}^n \sum_{t=1}^T f_j(x^{(t)}(i)) \). Following from Theorem 1, we obtain
\[ \Psi(t) = \mathbb{E}[\bar{F}_j(x(t))] \mathcal{H}_{t-1} = \mathbb{F}(x(t)) \leq \frac{Q}{t+1}, \] (67)

where
\[ Q = \max \{ C_i, C_i \}, \quad C_i = (9/2)(LD + 2DU_1 + 2LP_{\max}U_1) \] (68)

and
\[ C_i = \left( \frac{2p^2}{D} - 9 \right) \left( \frac{2p^2}{D^2} - \frac{16}{9} \frac{p^2}{D} \right) - \frac{1}{8} \] (69)

\[ U_{j_1} = \frac{n}{D} \left[ \frac{\bar{F}_j(x(t))}{D} - \frac{2\bar{F}_j(x(t))}{D} \right] \] (70)

\[ U'' = \frac{2L + 4L_{\max}U_1 + 4DU_2}{\mu_{\max}}. \] (71)

According to the definition of \( x^*(t) \), for any \( x^* \in \mathcal{X} \), we have
\[ \frac{1}{n} \sum_{i=1}^n \sum_{t=1}^T f_j(x(t)) \leq \frac{1}{n} \sum_{i=1}^n \sum_{t=1}^T f_j(x^*(i)). \] (72)

Due to the fact that \( x(t) = \arg \min_{x \in \mathcal{X}} \bar{F}_j(x) \) and \( \bar{F}_j(x) \) is the \( L/D \)-strong convex function, we have
\[ \mathbb{E}[\bar{F}_j(x(t))] \mathcal{H}_{t-1} - \bar{F}_j(x(t)) \geq \frac{L}{D} \mathbb{E}[\| x(t) - x^*(t) \|_2^2] \mathcal{H}_{t-1}. \] (73)

According to (67) and (73), we obtain
\[ \mathbb{E}[\| x(t) - x^*(t) \|_2^2] \mathcal{H}_{t-1} \leq \sqrt{\frac{D \Psi(t)}{L}} \leq \sqrt{\frac{QD}{L}} \frac{1}{t+1}. \] (74)

Since \( f_{j_1} \) is \( 3L \)-Lipschitz function, when \( t \geq t' \), the following inequality is obtained:
\[
\frac{1}{n} \sum_{i=1}^{n} E[f_i(x(t))] - \frac{1}{n} \sum_{i=1}^{n} f_i(x^*) \leq \frac{1}{\sqrt{T} + 1}.
\] (75)

Following the inequality
\[
\sum_{t=1}^{T} (1/\sqrt{t + 1}) \leq \int_{0}^{T+1} (1/\sqrt{t}) \, dt = 2\sqrt{T + 1}
\] (76)

and (75), we attain
\[
\frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} E[f_i(x(t))] - \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} f_i(x^*) \leq 6\sqrt{nLD}\sqrt{T + 1}.
\] (77)

By utilising
\[
-(1/n) \sum_{i=1}^{n} \sum_{t=1}^{T} (L/D) \|x^* - \bar{x}(t)\|_2 \leq 0
\] (78)

and (77), we obtain the following inequality:
\[
\frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} (\nabla f_i(x(t)), \bar{x}(t) - x^*) \leq 6\sqrt{nLD}\sqrt{T + 1}.
\] (79)

Since \( f_i \) is a convex function, we attain
\[
\frac{1}{n} \sum_{i=1}^{n} E[f_i(x(t))] - \frac{1}{n} \sum_{i=1}^{n} f_i(x^*) \leq \frac{1}{n} \sum_{i=1}^{n} (\nabla f_i(x(t)), \bar{x}(t) - x^*).\] (80)

Following from (79) and (80), we attain
\[
\frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} E[f_i(x(t))] - \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} f_i(x^*) \leq 6nLD\sqrt{T + 1}.
\] (81)

Hence, we obtain the final result of Theorem 2. □

6 Simulation experiments

To validate the results of our convergence analysis, we present some groups of simulation experiments in this section. The simulation experiments are executed on the public data set, aloi (http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/), for classification task. Moreover, aloi is a classification data set which contains 108,000 samples and 1000 classes. We randomly select 100 samples from aloi as input samples for each iteration. Therefore, the total number of iterations for one training process is 1000.

In the first group of our simulation experiments, we execute the proposed algorithm on the complete graph topology with different nodes, i.e. 1 node, 4 nodes, 64 nodes and 128 nodes. The results of this group simulation experiments are shown in Fig. 2. In addition, the average loss of the proposed algorithm decreases with iterations for all different nodes, which validates the convergence of the proposed algorithm.

In the second group of simulation experiments, we compare the convergence rate of the proposed algorithm on different topology graphs, i.e. complete graph, Watts–Strogatz graph and cycle graph. The results of this group simulation experiments are shown in Fig. 3. The results in Fig. 3 verify that the proposed algorithm is convergent on these three topology graphs.
Yi, P., Hong, Y., Liu, F.: ‘Distributed gradient algorithm for constrained optimization’. IEEE Trans. Inf. Theory, 2012, 58, (6), pp. 3575–3605

Beck, A., Nedić, A., Ozdaglar, A., et al.: ‘An o(1/k) gradient method for network resource allocation problems’. IEEE Trans. Control Netw. Syst., 2014, 1, (1), pp. 64–73

Xiao, L., Boyd, S.: ‘Optimal scaling of a gradient method for distributed resource allocation’. J. Optim. Theory Appl., 2006, 129, (3), pp. 469–488

Yi, P., Hong, Y., Liu, F.: ‘Distributed gradient algorithm for constrained optimization with application to load sharing in power systems’. Syst. Control Lett., 2015, 83, pp. 45–52

Namerikawa, T., Kato, T.: ‘Distributed load frequency control of electrical power networks via iterative gradient methods’. 2011 50th IEEE Conf. on Decision and Control and European Control Conf., Orlando, USA., 2011, pp. 7723–7728

Zhang, H., Quan, W., Chao, H.-C., et al.: ‘Smart identifier network: a collaborative architecture for the future internet’. IEEE Netw., 2016, 30, (3), pp. 46–51

Olfati-Saber, R., Alex Fax, J., Murray, R.M.: ‘Consensus and cooperation in networked multi-agent systems’. Proc. IEEE, 2007, 95, (1), pp. 215–233

Scutari, G., Sun, Y.: ‘Parallel and distributed successive convex approximation methods for big-data optimization’. 2018, pp. 141–308

Nedić, A., Olheiser, A.: ‘Distributed optimization over time-varying directed graphs’. IEEE Trans. Autom. Control, 2014, 60, (3), pp. 601–615

Nedić, A., Olshevsky, A., et al.: ‘Distributed subgradient methods for multi-agent optimization’. IEEE Trans. Autom. Control, 2009, 54, (1), pp. 48–61

Sundhar Ram, S., Nedić, A., Veeravalli, V.V.: ‘Distributed stochastic subgradient projection algorithms for convex optimization’. J. Optim. Theory Appl., 2010, 147, (3), pp. 516–545

Nedić, A., Lee, S., Raginsky, M.: ‘Decentralized online optimization with global objectives and local communication’. 2015 American Control Conf. (ACC), Chicago, USA., 2015, pp. 4497–4503

Hosseini, S., Chapman, A., Mesbahi, M.: ‘Online distributed optimization via dual averaging’. 52nd IEEE Conf. on Decision and Control, Florence, Italy, 2013, pp. 1484–1489

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., 2012, 25, (11), pp. 2483–2493

Hosseini, S., Chapman, A., Mesbahi, M.: ‘Online distributed convex optimization on dynamic networks’. IEEE Trans. Autom. Control, 2016, 61, (11), pp. 3545–3550

Akbari, M., Gharesifard, B., Linder, T.: ‘Distributed online convex optimization on time-varying directed graphs’. IEEE Trans. Control Netw. Syst., 2015, 4, (3), pp. 417–428

Xu, C., Zhu, J., Wu, D.O.: ‘Decentralized online learning methods based on weight-balancing over time-varying digraphs’. IEEE Trans. Emerg. Top. Comput. Intell., 2018

Frank, M., Wolfe, P.: ‘An algorithm for quadratic programming’. Nav. Res. Logist. Q., 1956, 3 (1-2), pp. 95–110

Zhang, W., Zhao, P., Zhu, W., et al.: ‘Projection-free distributed online learning in networks’. Proc. of the 34th Int. Conf. on Machine Learning, Sydney, Australia, 2017, pp. 4045–4046

Hazar, E., Agarwal, A., Kale, S.: ‘Logarithmic regret algorithms for online convex optimization’. Mach. Learn., 2007, 69, (2-3), pp. 169–192

Zhang, M., Quan, W., Cheng, N., et al.: ‘Distributed conditional gradient online learning for IOT optimization’. IEEE Internet Things J., 2019

Lacoste-Julien, S., Jaggi, M., Schmidt, M., et al.: ‘Block-coordinate Frank-Wolfe optimization for structural SVMs’. arXiv preprint arXiv:1207.4747, 2012

Zhang, L., Wang, G., Romero, D., et al.: ‘Randomized block Frank-Wolfe for convergent large-scale learning’. IEEE Trans. Signal Process., 2017, 65, (24), pp. 6448–6461

Wang, Y.-X., Sridharan, V., Dai, W., et al.: ‘Parallel and distributed block-coordinate Frank-Wolfe algorithms’. Proc. of the 33rd Int. Conf. on Machine Learning, New York, USA., 2016, pp. 1548–1557

Lasdon, J., Wei, H.-T., Modines, E.: ‘On the online Frank-Wolfe algorithms for convex and non-convex optimizations’. arXiv preprint arXiv:1510.01171, 2015