Online Learning over a Decentralized Network Through ADMM

Hao-Feng Xu, Qing Ling & Alejandro Ribeiro

Journal of the Operations Research Society of China

ISSN 2194-668X
Volume 3
Number 4

J. Oper. Res. Soc. China (2016) 3:537-562
DOI 10.1007/s40305-015-0104-0

Springer

J. Oper. Res. Soc. China (2015) 3:537–562
DOI 10.1007/s40305-015-0104-0

Online Learning over a Decentralized Network Through ADMM

Hao-Feng Xu[1] · **Qing Ling**[1] · **Alejandro Ribeiro**[2]

Received: 7 June 2015 / Revised: 12 October 2015 / Accepted: 29 October 2015 /
Published online: 7 December 2015

Abstract In this paper, we focus on the decentralized online learning problem where online data streams are separately collected and collaboratively processed by a network of decentralized agents. Comparing to centralized batch learning, such a framework is faithful to the decentralized and online natures of the data streams. We propose an online decentralized alternating direction method of multipliers that efficiently solves the online learning problem over a decentralized network. We prove its $O(\sqrt{T})$ regret bound when the instantaneous local cost functions are convex, and its $O(\log T)$ regret bound when the instantaneous local cost functions are strongly convex, where T is the number of iterations. Both regret bounds are in the same orders as those of centralized online learning. Numerical experiments on decentralized online least squares and classification problems demonstrate effectiveness of the proposed algorithm.

Keywords Multi-agent network · Decentralized optimization · Online learning

Mathematics Subject Classification 49R99 · 90C90

The work was supported by the National Natural Science Foundation of China (No. 61573331).

✉ Qing Ling
qingling@mail.ustc.edu.cn

Hao-Feng Xu
Haofeng@mail.ustc.edu.cn

Alejandro Ribeiro
aribeiro@seas.upenn.edu

[1] Department of Automation, University of Science and Technology of China, Hefei 230026, China

[2] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, USA

1 Introduction

This paper considers solving a decentralized online learning problem in a connected network composed of n agents. Let $\tilde{x} \in \mathbb{R}^p$ be a vector to learn. At time $k+1$, agent i receives an instantaneous local cost function $f_k^i(\tilde{x}) : \mathbb{R}^p \to \mathbb{R}$. Then agent i updates its estimate on $\tilde{x}$, which is denoted by x_{k+1}^i, based on the instantaneous local cost function $f_k^i(\tilde{x})$, its previous estimate x_k^i, as well as its neighbors' previous estimates x_k^j for all $j \in \mathcal{N}_i$ where $\mathcal{N}_i$ is the set of agent i's neighbors. After T iterations, the agents expect to find a network-wide optimal solution

$$\tilde{x}^* = \operatorname*{argmin}_{\tilde{x}} \sum_{k=1}^{T} \sum_{i=1}^{n} f_k^i(\tilde{x}). \tag{1.1}$$

When $n = 1$, the decentralized online learning problem (1.1) reduces to its centralized counterpart that finds applications in sequential decision making, e.g., online regression and classification [1]. If the online data streams are separately collected by decentralized agents and communications from the agents to a fusion center are costly, decentralized computing is a natural choice. Such application scenarios include wireless sensor networks and autonomous robot teams. Indeed, decentralized batch optimization in a multi-agent network has received extensive research interests recently [2–6]. This paper combines online learning and decentralized optimization and aims at designing an algorithm that is faithful to the decentralized and online natures of the data streams.

Observe that there exists a gap between the instantaneous local estimates and the optimal solution, which leads to the regret of online learning. In decentralized online learning, we encounter two types of regrets. The first is the nominal regret

$$\mathcal{R}_{\mathrm{N}} = \sum_{k=1}^{T} \sum_{i=1}^{n} \left(f_k^i \left(x_k^i \right) - f_k^i(\tilde{x}^*) \right), \tag{1.2}$$

which sums up the local regrets over the entire network. However, the nominal regret is unable to reflect the similarity of local estimates; it is possible that the local estimates are quite different but the nominal regret is small. Therefore, we introduce the second type of regret, termed as the global regret

$$\mathcal{R}_{\mathrm{G}} := \max_j \mathcal{R}_{\mathrm{G}}^j := \max_j \sum_{k=1}^{T} \sum_{i=1}^{n} \left(f_k^i \left(x_k^j \right) - f_k^i(\tilde{x}^*) \right). \tag{1.3}$$

Since any local estimate can be used as a final solution of decentralized online learning, we define $\mathcal{R}_{\mathrm{G}}$ as the maximum value of the aggregated regrets $\mathcal{R}_{\mathrm{G}}^j$ using all possible solutions x_k^j. Therefore, the global regret measures the quality of the local estimates from the network perspective. Different to the case of centralized online learning (i.e., $n = 1$) where $\mathcal{R}_{\mathrm{N}}$ and $\mathcal{R}_{\mathrm{G}}$ are identical, characterizing upper bounds of the global regret $\mathcal{R}_{\mathrm{G}}$ is of practical importance in decentralized online learning.

Here, we briefly introduce several existing decentralized online learning algorithms and their centralized counterparts, as well as the corresponding regret bounds. Therein, a common assumption is that the local or global cost functions have bounded subgradients. Motivated by the distributed subgradient method [3], in the distributed autonomous online learning algorithm, each agent combines neighboring estimates and descends along with the local subgradient direction [7]. When the local cost functions are convex, [7] proves an $O(\sqrt{T})$ regret bound that matches the centralized case in [8]. When the local cost functions are strongly convex, the regret bound is $O(\log T)$, the same order as that of the centralized online subgradient method [9]. Reference [10] develops an online version of the distributed dual averaging algorithm in [4] and proves an $O(\sqrt{T})$ regret bound with respect to the running-average local estimates for convex local cost functions. For strongly convex local cost functions, its regret bound is $O(\log T)$ in the running-average sense [11]. These bounds are similar to those established for centralized dual averaging in [12]. Reference [13] reformulates the decentralized online learning problem to a constrained form and approximately solves it with the saddle-point algorithm in [14] at every time. The algorithm alternates between primal subgradient descent and dual subgradient ascent, and the regret bound is $O(\sqrt{T})$ for the convex case.

All the aforementioned decentralized online learning algorithms are based on subgradient information of local cost functions, which fits for agents with limited computation abilities. If computation is not an issue, one can expect that letting each agent solve an optimization problem, other than move a subgradient descent step, yields a more powerful algorithm. This intuition motivates us to develop an online and decentralized version of the alternating direction method of multipliers (ADMM), i.e. the online Decentralized alternating direction Method of multipliers, named as oDM in this paper.

For decentralized batch optimization, ADMM first reformulates an optimization problem with consensus constraints, and then alternatingly minimizes its augmented Lagrangian function with respect to two blocks of variables linked by the consensus constraints and updates the Lagrange multipliers. The resulting algorithm is fully decentralized where each agent minimizes the sum of its local cost function and a time-varying quadratic term [2], followed by the update of its local Lagrange multiplier. The decentralized ADMM has found successful applications in wireless sensor networks, computer networks, smart grids, etc [15–18]. Analysis of its convergence follows that of the centralized ADMM [19,20], and its linear rate of convergence is established in [21]. The decentralized ADMM is also able to handle the case that the local cost functions are dynamic [22].

This paper is related to the centralized online ADMM algorithm and its regret analysis in [23]; however, the proposed oDM is designed from a different perspective and its analysis is along a different line as explained below.

The centralized online ADMM minimizes an instantaneous cost function that is the sum of two separable functions with respect to two blocks of variables; the two blocks of variables are linked with linear constraints. Through alternating minimization of the augmented Lagrangian function and update of the Lagrange multipliers, the centralized online ADMM is able to handle the constrained online learning problem. In this paper, we consider the unconstrained decentralized online learning prob-

lem (1.1) that is reformulated to a constrained one through introducing a block of auxiliary variables and linking the auxiliary variables with the local estimates via consensus constraints (see Sect. 2). Applying ADMM to this online constrained formulation leads to a decentralized algorithm (see Sect. 3). In other words, the technique of ADMM is used to handle the intrinsic constraints in the centralized online algorithm, while it makes the iterations decentralized in the proposed algorithm.

Further, the regret derived in [23] contains two parts, one is about function value, which corresponds to the nominal regret in the decentralized regime, and another is the sum of the norms of constraint violations over time. The regret with respect to function value is $O(\sqrt{T})$ for the convex case and $O(\log T)$ for the strongly convex case under the assumption of bounded subgradients. The regret with respect to constraint violation is also $O(\sqrt{T})$ for the convex case and $O(\log T)$ for the strongly convex case. Since the regret with respect to constraint violation does not translate to the gap between the global and nominal regrets, implanting the analysis of [23] to the decentralized case cannot directly yield the same bounds for the global regret. In this paper, we establish the gap between the global and nominal regrets through a series of supporting lemmas (see Lemmas 4.2–4.5 in Sect. 4.1). Based on these lemmas, we prove that when the local cost functions are convex and have bounded subgradients, the global regret $\mathcal{R}_{\mathrm{G}}$ has an $O(\sqrt{T})$ bound (see Theorem 4.6 in Sect. 4.2). For the strongly convex case, the bound is $O(\log T)$ under the same assumption of bounded subgradients (see Theorem 4.7 in Sect. 4.2).

2 Problem Statement

Throughout this paper, we consider a network composed of a set of n agents $\mathcal{V} = \{1, 2, \cdots, n\}$ and a set of m arcs $\mathcal{A} = \{1, 2, \cdots, m\}$. Here each arc $a \sim (i, j)$ is associated with an ordered pair (i, j) indicating that i can communicate to j. We assume the network is connected and communication is bidirectional. The set of agents adjacent to i is termed as its neighborhood and denoted as $\mathcal{N}_i$. The cardinality of this set is the degree d_i of agent i. We define the block arc source matrix $A_{\mathrm{s}} \in \mathbb{R}^{mp} \times np$ where the block $(A_{\mathrm{s}})_{a,i} = I_p \in \mathbb{R}^{p\times p}$ is an identity matrix if the arc $a \sim (i, j)$ originates at agent i and is null otherwise. Likewise, define the block arc destination matrix $A_{\mathrm{d}} \in \mathbb{R}^{mp\times np}$ where the block $(A_{\mathrm{d}})_{a,i} = I_p \in \mathbb{R}^{p\times p}$ if the arc $a \sim (i, j)$ terminates at the node j and is null otherwise. Observe that the extended oriented incidence matrix can be written as $E_{\mathrm{o}} = A_{\mathrm{s}} - A_{\mathrm{d}}$ and the unoriented incidence matrix as $E_{\mathrm{u}} = A_{\mathrm{s}} + A_{\mathrm{d}}$. The extend oriented Laplacian is given by $L_{\mathrm{o}} = \frac{1}{2}E_{\mathrm{o}}^{\mathrm{T}}E_{\mathrm{o}}$, the unoriented Laplacian is $L_{\mathrm{u}} = \frac{1}{2}E_{\mathrm{u}}^{\mathrm{T}}E_{\mathrm{u}}$, and the degree matrix containing nodes' degrees d_i in the diagonal is $D = \frac{1}{2}(L_{\mathrm{o}} + L_{\mathrm{u}})$. The largest eigenvalue of the unoriented Laplacian is denoted by Γ_{u}, which is related to the connectedness of the network; smaller Γ_{u} means better connectedness.

At time $k+1$, decentralized online learning (approximately) solves an optimization problem

$$\{x_{k+1}^i\} = \underset{\{x^i\}}{\operatorname{argmin}} \sum_{i=1}^{n} f_k^i(x^i) + \frac{\eta_{k+1}}{2} \sum_{i=1}^{n} \|x^i - x_k^i\|^2 \quad \text{s.t. } x^1 = x^2 = \cdots = x^n. \tag{2.1}$$

In (2.1), $\sum_{i=1}^n f_k^i(x^i)$ is the summation of the local cost functions f_k^i evaluated at different local estimates x^i, η_{k+1} is a constant or time-varying positive parameter such that the term $(\eta_{k+1}/2)\sum_{i=1}^n \|x^i - x_k^i\|^2$ makes the local estimates x^i close to their previous values x_k^i, and the constraints $x^1 = x^2 = \cdots = x^n$ enforce all the local estimate x^i to reach a network-wide consensus. Observe that the consensus constraints couple all the agents in the entire network and are thus nontrivial.

To address this issue, we introduce auxiliary variables $z^{ij} \in \mathbb{R}^p$ associated with arcs $(i, j) \in \mathcal{A}$ and rewrite (2.1) as

$$\{x_{k+1}^i, z_{k+1}^{ij}\} = \underset{\{x^i, z^{ij}\}}{\operatorname{argmin}} \sum_{i=1}^{n} f_k^i(x^i) + \frac{\eta_{k+1}}{2} \sum_{i=1}^{n} \|x^i - x_k^i\|^2, \quad \text{s.t. } x^i = z^{ij},\ x^j = z^{ij},\quad \forall\, (i, j) \in \mathcal{A}. \tag{2.2}$$

The constraints $x^i = z^{ij}$ and $x^j = z^{ij}$ imply that for all pairs of agents $(i, j) \in \mathcal{A}$ forming arcs, the feasible set of (2.2) is such that $x^i = x^j$. We interpret the auxiliary variables z^{ij} as being attached to the arc (i, j) with the purpose of enforcing the equality of the variables x^i and x^j attached to its source agent i and destination agent j. For a connected network, these local neighborhood constraints further imply that feasible variables must satisfy $x^i = x^j$ for all, not necessarily neighboring, pairs of agents i and j. As a consequence, the optimal local variables x^i in (2.2) must coincide with those in (2.1).

For clarity of discussion, we define $x = [x^1; x^2; \cdots; x^n] \in \mathbb{R}^{np}$ that containing all variables x^i and $z = [z^1; z^2; \cdots; z^m] \in \mathbb{R}^m p$ that containing all variables $z^a = z^{ij}$ if $a \sim (i, j)$. Recalling the definitions of the arc source matrix A_s and the arc destination matrix A_d, the consensus constraints (2.2) can be represented as $A_s x - z = 0$ and $A_d x - z = 0$. We further define the aggregated function $f_k : \mathbb{R}^{np} \to \mathbb{R}$ as $f_k(x) := \sum_{i=1}^n f_k^i(x^i)$. Using these definitions, we can rewrite (2.2) in a matrix form as

$$\{x_{k+1}, z_{k+1}\} = \underset{\{x,z\}}{\operatorname{argmin}}\ f_k(x) + \frac{\eta_{k+1}}{2}\|x - x_k\|^2 \quad \text{s.t. } Ax + Bz = 0, \tag{2.3}$$

where $A = [A_s; A_d]$, $B = [-I_{mp}; -I_{mp}]$.

Given time $k + 1$, (2.3) fits the standard form that can be solved by ADMM, i.e., minimizing the sum of two separable functions $f_k(x) + (\eta_{k+1}/2)\|x - x_k\|^2$ and 0 with respect to two blocks of variables x and z under the linear constraint $Ax + Bz = 0$. Based on this reformulation, we develop a decentralized online algorithm that utilizes the idea of ADMM.

3 Algorithm Development

This section develops oDM that sequentially and approximately solves (2.3) (and (2.1), equivalently). Associate the constraint $A_s x - z = 0$ with a Lagrange multiplier $\beta \in \mathbb{R}^m p$ and $A_d x - z = 0$ with $\gamma \in \mathbb{R}^m p$, and denote $\lambda = [\beta; \gamma] \in \mathbb{R}^{2mp}$ as the Lagrange multiplier associated with the constraint $Ax + Bz = 0$. At any time $k+1$, the augmented Lagrangian function of (2.3) is

$$L_{k+1}(x, z, \lambda) = f_k(x) + \frac{\eta_{k+1}}{2}\|x - x_k\|^2 + \lambda^{\mathrm{T}}(Ax + Bz) + \frac{\rho_{k+1}}{2}\|Ax + Bz\|^2, \quad (3.1)$$

where the stepsize ρ_{k+1} is a constant or time-varying positive value.

In oDM, the augmented Lagrangian function $L_{k+1}(x, z, \lambda)$ is minimized with respect to the primal variables x and z in an alternating manner, followed by an ascent step on the dual variable λ. To be specific, given past iterates z_k and λ_k, the primal iterate x_{k+1} is defined as $x_{k+1} := \mathrm{argmin}_x L_{k+1}(x, z_k, \lambda_k)$ and given as the solution of the first-order optimality condition

$$\partial f_k(x_{k+1}) + A^{\mathrm{T}}\lambda_k + \rho_{k+1}A^{\mathrm{T}}(Ax_{k+1} + Bz_k) + \eta_{k+1}(x_{k+1} - x_k) \ni 0. \quad (3.2)$$

Using the value of x_{k+1} from (3.2) along with the previous dual iterate λ_k, the primal iterate z_{k+1} is defined as $z_{k+1} := \mathrm{argmin}_z L_{k+1}(x_{k+1}, z, \lambda_k)$ and explicitly given by the solution of the first-order optimality condition

$$B^{\mathrm{T}}\lambda_k + \rho_{k+1}B^{\mathrm{T}}(Ax_{k+1} + Bz_{k+1}) = 0. \quad (3.3)$$

The dual iterate λ_k is then updated by the constraint violation $Ax_{k+1} + Bz_{k+1}$ corresponding to primal iterates x_{k+1} and z_{k+1} such that

$$\lambda_{k+1} - \lambda_k - \rho_{k+1}(Ax_{k+1} + Bz_{k+1}) = 0. \quad (3.4)$$

Note that in the online learning setting, iteration $k+1$ runs (3.2)–(3.4) once, and hence (2.3) is only approximately solved.

The computations to implement (3.2)–(3.4) are decentralized through the network. However, it is also possible to rearrange (3.2)–(3.4) such that with proper initialization the updates of the auxiliary variables z_k are not necessary, and the Lagrange multipliers $\beta \in \mathbb{R}^{mp}$ and $\gamma \in \mathbb{R}^{mp}$ can be replaced by a smaller dimension vector $\alpha = [\alpha^1; \alpha^2; \cdots; \alpha^n] \in \mathbb{R}^{np}$. We summarize the initialization conditions and the simplified algorithms in the following proposition.

Proposition 3.1 *Consider the sequence of variables x_{k+1} generated by (3.2)–(3.4). If the algorithm is initialized with $\beta_1 = -\gamma_1$ and $\frac{1}{2}E_u x_1 = z_1$, then the iterates x_{k+1} can be alternatively generated by the recursion*

$$\partial f_k(x_{k+1}) + \alpha_k + 2\rho_{k+1}Dx_{k+1} - \rho_{k+1}L_u x_k + \eta_{k+1}(x_{k+1} - x_k) \ni 0, \quad (3.5)$$

$$\alpha_{k+1} - \alpha_k - \rho_{k+1}L_o x_{k+1} = 0. \quad (3.6)$$

Proof See Appendix 1.

The updates of (3.5) and (3.6) are decentralized to the agents. Indeed, using the definitions of the degree matrix D, the unoriented Laplacian L_{u}, and the oriented Laplacian L_{o}, the iterations are equivalent to

$$\partial f_k^i\left(x_{k+1}^i\right) + \left(\eta_{k+1} + 2\rho_{k+1}d_i\right) x_{k+1}^i - \left(\eta_{k+1}x_k^i + \rho_{k+1}\sum_{j\in\mathcal{N}_i}\left(x_k^i + x_k^j\right)\right) + \alpha_k^i \ni 0, \tag{3.7}$$

$$\alpha_{k+1}^i = \alpha_k^i + \rho_{k+1}\sum_{j\in\mathcal{N}_i}\left(x_{k+1}^i - x_{k+1}^j\right). \tag{3.8}$$

The update of x_{k+1}^i in (3.7) and the update of α_{k+1}^i in (3.8) are decentralized since they only rely on local and neighboring information. We summarize oDM in Algorithm 1.

Algorithm 1 oDM at agent i

Require: Initialize local variables to $x_1^i = 0$ and $\alpha_1^i = 0$.

Require: Initialize neighboring variables $x_1^j = 0$ for all $j \in \mathcal{N}_i$.

Step 1. **for** $k + 1 = 2, 3, \cdots$ **do**

Step 2. Observe instantaneous local cost function f_k^i.

Step 3. Set parameters $\rho_{k+1} > 0$ and $\eta_{k+1} > 0$.

Step 4. Compute local estimate x_{k+1}^i from [cf. (3.7)]

$$\partial f_k^i\left(x_{k+1}^i\right) + \left(\eta_{k+1} + 2\rho_{k+1}d_i\right) x_{k+1}^i - \left(\eta_{k+1}x_k^i + \rho_{k+1}\sum_{j\in\mathcal{N}_i}\left(x_k^i + x_k^j\right)\right) + \alpha_k^i \ni 0.$$

Step 5. Transmit x_{k+1}^i to and receive x_{k+1}^j from neighbors $j \in \mathcal{N}_i$.

Step 6. Update local variable α_{k+1}^i as [cf. (3.8)]

$$\alpha_{k+1}^i = \alpha_k^i + \rho_{k+1}\sum_{j\in\mathcal{N}_i}\left(x_{k+1}^i - x_{k+1}^j\right).$$

Step 7. **end for**

4 Regret Bounds

This section analyzes the regret bounds of oDM for two cases: (1) an $O(\sqrt{T})$ regret bound when the local cost functions f_k^i are convex; (2) an $O(\log T)$ regret bound when the local cost functions f_k^i are strongly convex. The roadmap of the analysis is as following. First, we show that the global regret $\mathcal{R}_{\mathrm{G}}$ can be upper bounded by the summation of three terms: the nominal regret $\mathcal{R}_{\mathrm{N}}$, the accumulated constraint violation, and the accumulated inverse of stepsize (see Sect. 4.1). Second, we prove

that the summation of the first two terms (the nominal regret $\mathcal{R}_{\mathrm{N}}$ and the accumulated constraint violation) and the third term (the accumulated inverse of stepsize) both have the $O(\sqrt{T})$ and $O(\log T)$ regret bounds for the convex and strongly convex cases, respectively (see Sect. 4.2).

Throughout the analysis, we assume that a finite optimal solution $\tilde{x}^*$ to (1.1) exists. Define x^* as a vector stacking n copies of $\tilde{x}^*$ and z^* as a vector stacking m copies of $\tilde{x}^*$. Apparently, x^* and z^* satisfy the equality $Ax^* + Bz^* = 0$. Assume that $\|\tilde{x}^*\| \leqslant D_{\tilde{x}}$, which implies that $\|x^*\| \leqslant \sqrt{n} D_{\tilde{x}}$, $\|z^*\| \leqslant \sqrt{m} D_{\tilde{x}}$, and $\|Bz^*\| \leqslant \sqrt{2m} D_{\tilde{x}}$.

We also assume that the local cost functions have bounded subgradients. This assumption is common in the centralized/decentralized online learning literature.

Assumption 4.1 (Bounded Subgradient) For all agents i and all times k, the subgradients of the local cost functions ∂f_k^i with respect to the Euclidean norm are upper bounded by a positive constant L_{f}. In consequence for all times k, the aggregated cost functions f_k satisfy $\|\partial f_k\| \leqslant \sqrt{n} L_{\mathrm{f}}$.

4.1 Supporting Lemmas

This subsection proves a series of supporting lemmas and shows that the global regret $\mathcal{R}_{\mathrm{G}}$ contains three components: the nominal regret $\mathcal{R}_{\mathrm{N}}$, the accumulated constraint violation, and the accumulated inverse of stepsize. The discussion is based on Lemma 4.2.

Lemma 4.2 *Consider the points (x_k, z_k, λ_k) generated by the iterations (3.2)–(3.4). Recall that the stepsize at time k is ρ_k and define the diameter of the network of n agents as σ. If the local cost functions f_k^i are convex and Assumption 1 holds, then the global regret*

$$\mathcal{R}_{\mathrm{G}} \leqslant \mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T} \frac{\rho_k}{2n\sigma} \max_j \sum_{i=1}^{n} \|x_k^j - x_k^i\|^2 + \frac{n^2 \sigma L_{\mathrm{f}}^2}{2} \sum_{k=1}^{T} \frac{1}{\rho_k}. \tag{4.1}$$

Proof See Appendix 2.

Observe that given one local estimate x_k^j, $\sum_{i=1}^n \|x_k^j - x_k^i\|^2$ accumulates its discrepancies with others. Therefore, in (4.1) the second term $\max_j \sum_{i=1}^n \|x_k^j - x_k^i\|^2$ means the largest accumulated discrepancy over the entire network. Since $\|x_k^j - x_k^i\|^2$ does not explicitly appear in the oDM iterates, we replace it by the constraint violation term $\|Ax_k + Bz_k\|^2$ (see Lemma 4.3). Further using the fact that $\|Ax_{k+1} + Bz_{k+1}\|^2 \leqslant \|Ax_{k+1} + Bz_k\|^2$ (see Lemma 4.4), we can replace the second term of (4.1) by the constraint violation with respect to the primal solutions x_{k+1} and z_k, which enables the analysis of the regret bounds.

Lemma 4.3 *Consider the points (x_k, z_k, λ_k) generated by the iterations (3.2)–(3.4). If the length of the shortest path from agent i to agent j is σ_{ij} $(1 \leqslant \sigma_{ij} \leqslant n-1)$, then for all times k*

$$\|x_k^i - x_k^j\|^2 \leqslant \sigma_{ij} \|Ax_k + Bz_k\|^2. \tag{4.2}$$

Proof See Appendix 3.

Recalling that $\sigma = \max_{i,j} \sigma_{ij}$ is the diameter of the network, an immediate conclusion from Lemma 4.3 is that $\max_{i,j} \|x_k^i - x_k^j\|^2 \leqslant \sigma \|Ax_k + Bz_k\|^2$, which is determined by the constraint violation and the network topology. The following lemma further states that the constraint violation at time $k+1$ (i.e., $\|Ax_{k+1} + Bz_{k+1}\|^2$) is no greater than the constraint violation with respect to the primal solutions x_{k+1} and z_k (i.e., $\|Ax_{k+1} + Bz_k\|^2$).

Lemma 4.4 *Consider the points (x_k, z_k, λ_k) generated by the iterations (3.2)–(3.4) under the initialization $x_1 = 0$, $z_1 = 0$, and $\lambda_1 = 0$. For all times k, it holds*

$$\|Ax_{k+1} + Bz_{k+1}\|^2 \leqslant \|Ax_{k+1} + Bz_k\|^2. \tag{4.3}$$

Proof See Appendix 4.

According to Lemmas 4.2–4.4, the global regret $\mathcal{R}_G$ is upper bounded by the summation of three terms: the nominal regret $\mathcal{R}_N$, the accumulated constraint violation, and the accumulated inverse of stepsize. We conclude it with the lemma below.

Lemma 4.5 *Consider the points (x_k, z_k, λ_k) generated by the iterations (3.2)–(3.4) under the initialization $x_1 = 0$, $z_1 = 0$, and $\lambda_1 = 0$. Define the length of the shortest path from agent i to agent j as σ_{ij} ($1 \leqslant \sigma_{ij} \leqslant n-1$) and define the diameter of the network of n agents as $\sigma = \max_{i,j} \sigma_{ij}$. Then for all times k*

$$\mathcal{R}_G \leqslant \mathcal{R}_N + \sum_{k=1}^{T} \frac{\rho_{k+1}}{2} \|Ax_{k+1} + Bz_k\|^2 + \frac{n^2 \sigma L_f^2}{2} \sum_{k=1}^{T} \frac{1}{\rho_k}. \tag{4.4}$$

Proof See Appendix 5.

4.2 Bounds of the Global Regret $\mathcal{R}_G$

We first establish the bound of the global regret $\mathcal{R}_G$ given that the local cost functions f_k^i are convex.

Theorem 4.6 *Consider the points (x_k, z_k, λ_k) generated by the iterations (3.2)–(3.4) under the initialization $x_1 = 0$, $z_1 = 0$, and $\lambda_1 = 0$. The network has n agents and m arcs, and the network diameter is σ. If the local cost functions f_k^i are convex and Assumption 4.1 holds, letting the algorithm parameters $\rho_k = c_1 n \sqrt{T}$ and $\eta_k = \frac{L_f}{D_{\tilde{x}}} \sqrt{T}$ where $c_1 > 0$ is an arbitrary constant, then the global regret satisfies*

$$\mathcal{R}_G \leqslant \left(c_1 n m D_{\tilde{x}}^2 + n L_f D_{\tilde{x}} + \frac{n \sigma L_f^2}{2 c_1} \right) \sqrt{T}. \tag{4.5}$$

Proof See Appendix 6.

Remark We briefly discuss how to choose the arbitrary constant c_1 such as the coefficient $c_1 nm D_{\tilde{x}}^2 + nL_f D_{\tilde{x}} + (n\sigma L_f^2)/(2c_1)$ is order optimal with respect to n, the number of the agents. Observe that for the two topology-related constants m and σ, the number of arcs $m = O(n)$ in sparse topologies and $m = O(n^2)$ in dense topologies; the network diameter $\sigma = O(n)$ in line-like topologies, $\sigma = O(\sqrt{n})$ in grid-like topologies, and $\sigma = O(1)$ in dense topologies. Therefore, for dense topologies ($m = O(n^2)$ and $\sigma = O(1)$), choosing $c_1 = O(1/n)$ yields a coefficient at the order of $O(n^2)$. For line-like topologies ($m = O(n)$ and $\sigma = O(n)$), $c_1 = O(1)$ also yields a coefficient at the order of $O(n^2)$. For grid-like topologies ($m = O(n)$ and $\sigma = O(\sqrt{n})$), the order improves to $O(n^{7/4})$ by choosing $c_1 = O(n^{-1/4})$. The best is for the star-like topologies ($m = O(n)$ and $\sigma = O(1)$), where the order is $O(n^{3/2})$ by choosing $c_1 = O(n^{-1/2})$.

Next we further establish the bound of the global regret $\mathcal{R}_G$ given that the local cost functions f_k^i are strongly convex.

Theorem 4.7 *Consider the points (x_k, z_k, λ_k) generated by the iterations (3.2)–(3.4) under the initialization $x_1 = 0$, $z_1 = 0$, and $\lambda_1 = 0$. The network has n agents and m arcs, the network diameter is σ, and the largest eigenvalue of the unoriented Laplacian $L_u = \frac{1}{2} E_u^T E_u$ is Γ_u. If the local cost functions f_k^i are μ-strongly convex and Assumption 4.1 holds, letting the algorithm parameters $\rho_k = \frac{\mu}{2\Gamma_u} k$ and $\eta_k = \frac{\mu}{2} k$, then the global regret satisfies*

$$\mathcal{R}_G \leqslant \frac{2m + \Gamma_u n}{2\Gamma_u n} \mu D_x^2 + \left(\frac{nL_f^2}{\mu} + \frac{\Gamma_u n^2 \sigma L_f^2}{\mu} \right) (\log T + 1). \tag{4.6}$$

Proof See Appendix 7.

Observe that the $O(\sqrt{T})$ regret bound in Theorem 4.6 and the $O(\log T)$ regret bound in Theorem 4.7 match the optimal ones established in centralized online learning.

5 Numerical Experiments

This section demonstrates performance of oDM for decentralized online learning problems. Specifically, we consider a least squares problem and a classification problem. In the numerical experiments, we compare oDM with two decentralized online learning algorithms:

(1) Distributed online gradient descent (DOGD) [11];
(2) Distributed autonomous online learning (DAOL) [7].

Recall that DOGD is based on the distributed dual averaging algorithm and DAOL is an online version of the distributed subgradient method; see the paper survey in Sect. 1. For fair comparison, we hand-tune the algorithm parameters to the best ones.

5.1 Decentralized Online Least Squares

We consider the decentralized online least square problem. Suppose that a network of n agents take linear measurements on a vector $\tilde{x} \in \mathbb{R}^{10}$. For agent i at time k, the measurement is $b_k^i = A^i \tilde{x} + e_k^i \in \mathbb{R}^{10}$ where $A^i \in \mathbb{R}^{1\times 10}$ is the measurement matrix and $e_k^i \in \mathbb{R}^{10}$ is the measurement error. Elements of the true $\tilde{x}$ and A^i follow zero mean normal distribution with variance 1, while elements of e_k^i follow the zero mean normal distribution with variance 10^{-2}. In decentralized online least squares, the instantaneous local cost function of agent i at time k is $f_k^i(\tilde{x}) = (1/2)\|A^i\tilde{x} - b_k^i\|^2$. Given such f_k^i, the oDM iterations (3.7) and (3.8) become

$$\begin{aligned} x_{k+1}^i &= \left((A^i)^{\mathrm{T}} A^i + (\eta_{k+1} + 2\rho_{k+1} d_i) I_{10}\right)^{-1} \\ &\quad \times \left(\eta_{k+1} x_k^i + \rho_{k+1} \sum_{j\in\mathcal{N}_i} \left(x_k^i + x_k^j\right) + (A^i)^{\mathrm{T}} b_k^i - \alpha_k^i\right), \\ \alpha_{k+1}^i &= \alpha_k^i + \rho_{k+1} \sum_{j\in\mathcal{N}_i} \left(x_{k+1}^i - x_{k+1}^j\right). \end{aligned}$$

In the numerical experiments, we generate data from times 1 to $T = 200$. We define two performance metrics. One is the average loss

$$\frac{1}{nT} \sum_{t=1}^{T} \sum_{i=1}^{n} \left(f_{\mathrm{t}}^i\left(x_k^j\right) - f_{\mathrm{t}}^i(\tilde{x}^*)\right),$$

which, at time k, evaluates the local estimate x_k^j of an arbitrary agent j on all local cost functions and then averages over times 1 to T. Another is the average disagreement

$$\frac{1}{n} \sum_{i=1}^{n} \frac{\|x_k^j - x_k^i\|}{\|\tilde{x}^*\|},$$

which characterizes the disagreement between the instantaneous local estimates and the one held by an arbitrary reference agent j. For simplicity, we let $j = 1$ here.

In the first set of numerical experiments, we compare the three decentralized online learning algorithms on a network of $n = 100$ agents where 1 980 arcs out of 9 900 possible ones are randomly chosen to be connected. Average losses and average disagreements of the decentralized online algorithms are shown in Fig. 1. oDM demonstrates the fastest convergence among the three algorithms. DOGD performs well at the low-accuracy stage and degrades at the high-accuracy stage. Observe that DOGD keeps the tightest consensus during the learning process, while oDM has the largest disagreement. This observation suggests that keeping tight consensus during the learning process may be unfavorable since the consensual local solutions are possibly far away from the optimum, which yields worse average loss. Performance of DAOL is between those of oDM and DOGD, in terms of both average loss and average disagreement.

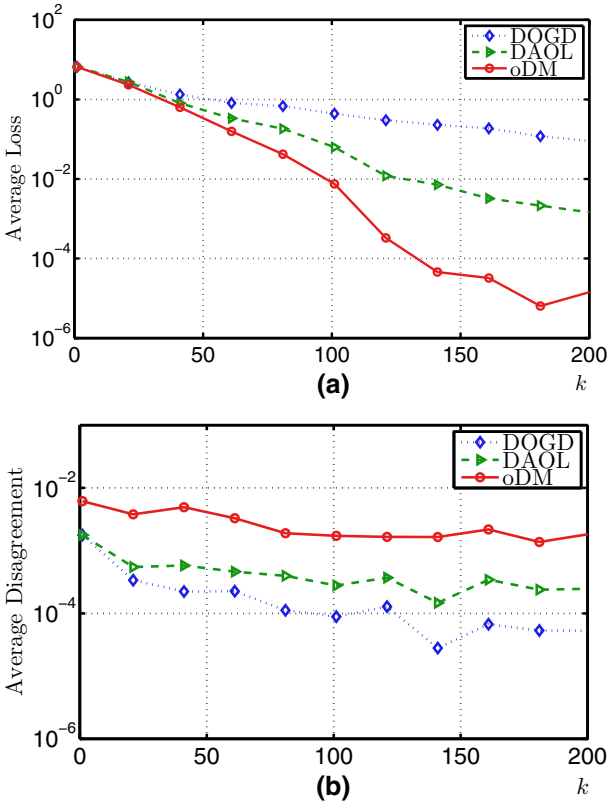

Fig. 1 Performance of the decentralized online learning algorithms on the least squares problem: (**a**) Average Loss; (**b**) Average Disagreement

In the second set of numerical experiments, we discuss the impact of topology on the performance of oDM. Consider networks of $n = 100$ agents where $9\,900 \times r$ random arcs are connected. The value of r determines the ratio of connected arcs; larger r means better network connectivity. Observe from Fig. 2 that better connectivity yields faster convergence and tighter consensus. However, the performance degradation from a dense network ($r = 0.8$) to a sparse network ($r = 0.05$) is not remarkable, suggesting that it is beneficial to use a sparse network when the communication cost is an issue. Similar observation holds for special topologies such as line, star, and complete, see Fig. 3. Unless the network is extremely sparse (line), oDM shows similar convergence properties.

In the third set of numerical experiments, we show the influence of network size on the performance of oDM. We choose different values of n and keep the ratio of connected arcs invariant; specifically, we let $n(n-1) \times 20\%$ random arcs be connected. Fig. 4 depicts convergence of oDM as the network size varies. Larger network size leads to sharper reduction of the average loss and the average disagreement; this makes sense since more data are involved in the learning process.

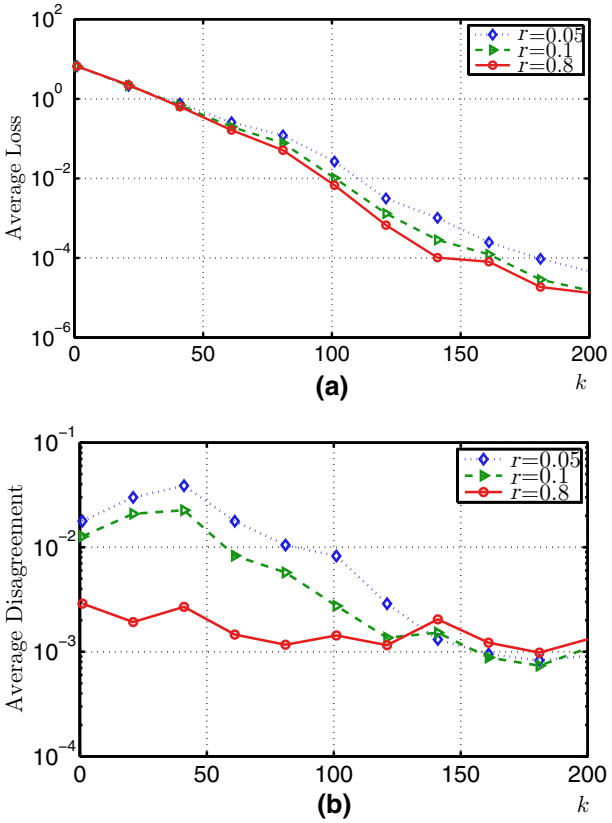

Fig. 2 Performance of oDM on the least squares problem with random networks: (**a**) Average Loss; (**b**) Average Disagreement

5.2 Decentralized Online Classification

Next we consider the decentralized online classification problem. We use 30 000 training samples and 16 281 testing samples from the a9a dataset [24]. The training samples are trained with a network of $n = 30$ agents in an online manner, and the time duration is $T = 1\,000$. At time k, agent i has a training sample (a_k^i, b_k^i) where $a_k^i \in \mathbb{R}^{123}$ is a vector containing 123 features and b_k^i is a binary-valued scalar representing the label.

The task of decentralized online learning is to train a classifier $\tilde{x} \in \mathbb{R}^{123}$. Here, we choose the support vector machine as the optimization model [25]. At time k, the instantaneous local cost function of agent i is $f_k^i(\tilde{x}) = (\kappa/2)\|\tilde{x}\|^2 + \max\{1 - b_k^i\langle a_k^i, \tilde{x}\rangle, 0\}$ where the regularization parameter $\kappa = 0.1$. Given such f_k^i, the oDM updates (3.7) and (3.8) become

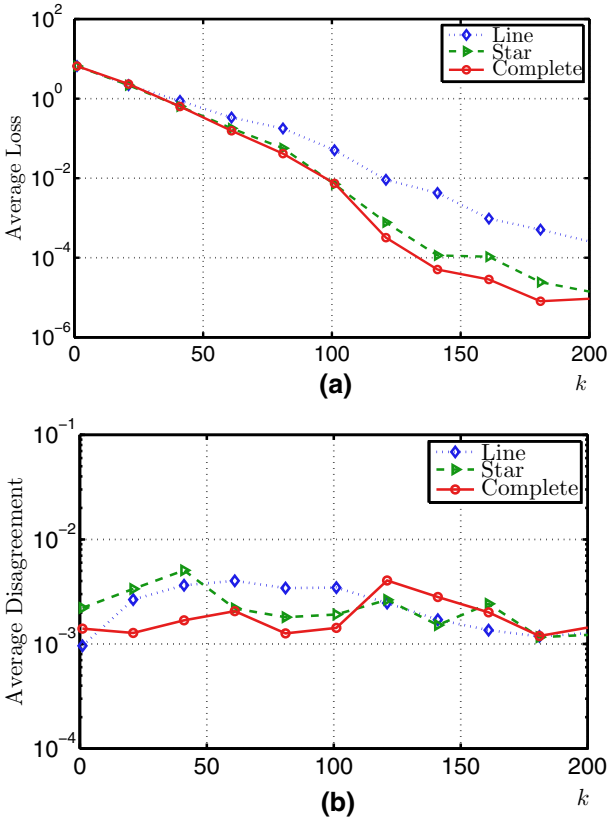

Fig. 3 Performance of oDM on the least squares problem with special topologies: (**a**) Average Loss; (**b**) Average Disagreement

$$x_{k+1}^i = \operatorname*{argmin}_{x^i} \frac{\kappa + \eta_{k+1} + 2\rho_{k+1} d_i}{2} \|x^i\|^2 + \max\{1 - b_{k+1}^i \langle a_{k+1}^i, x^i \rangle, 0\} + \left\langle \eta_{k+1} x_k^i + \rho_{k+1} \sum_{j \in \mathcal{N}_i} (x_k^i + x_k^j) - \alpha_k^i, x^i \right\rangle$$

$$\alpha_{k+1}^i = \alpha_k^i + \rho_{k+1} \sum_{j \in \mathcal{N}_i} \left(x_{k+1}^i - x_{k+1}^j \right).$$

We compare the three decentralized online learning algorithms on a network of $n = 30$ agents where 174 arcs out of 870 possible ones are randomly chosen to be connected. As a baseline result, we also use Libsvm to run centralized batch learning. The performance metric is error rate, which is defined as the rate of wrong classifications on the testing samples using an arbitrary instantaneous classifier. Fig. 5 shows error rates of the three decentralized online learning algorithms and the centralized Libsvm. oDM has the best performance and is close to the centralized Libsvm. The two

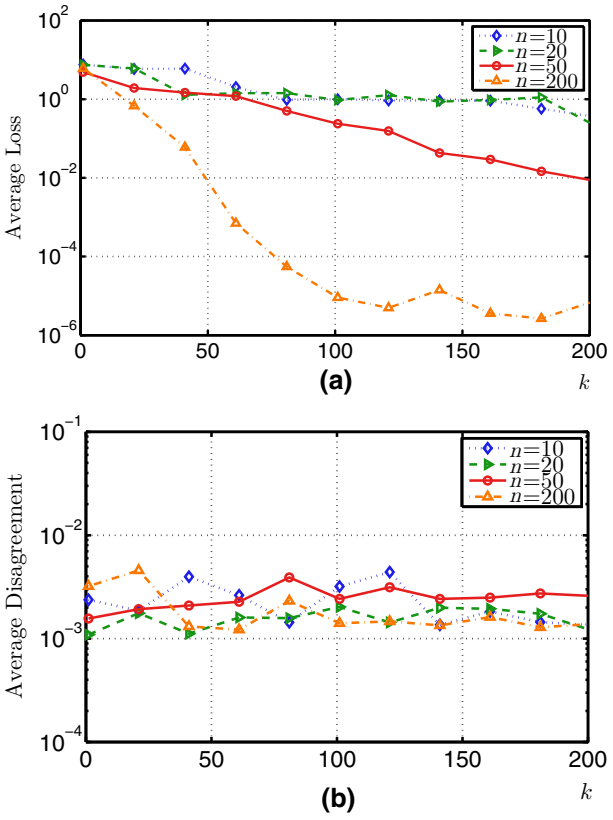

Fig. 4 Performance of oDM on the least squares problem with different network size n: (**a**) Average Loss; (**b**) Average Disagreement

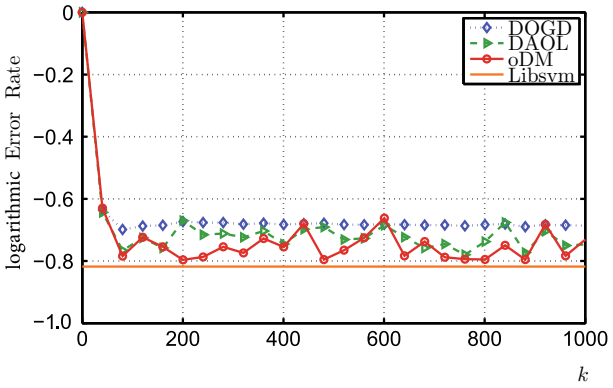

Fig. 5 Error rates of the decentralized online learning algorithms and the centralized Libsvm on the classification problem

other decentralized algorithms, DOGD and DAOL, demonstrate similar error rates, which are slightly worse than that of oDM.

6 Conclusion

This paper considers the machine learning application where the samples are not only online temporally but also decentralized in spatially. The decentralized online setting requires agents to incorporate new samples with current solutions, as well as cooperate with neighboring agents for network-wide consensus. We proposed the oDM to solve this problem; therein the alternating direction method of multipliers (ADMM) plays two roles, i.e., making the algorithm decentralized and enabling online computation. We define the regret bound for the decentralized online learning problem, and prove $O(\sqrt{T})$ and $O(\log T)$ regret bounds for oDM when the instantaneous local cost functions are convex and strongly convex, respectively. Numerical experiments on decentralized online least squares and classification problems demonstrate the effectiveness of the proposed algorithm.

Appendix 1: Proof of Proposition 3.1

Proof Multiplying the two sides of the λ-update (3.4) by A^{T} (or B^{T}) and adding it to the x-update (3.2) (or the z-update (3.3)), (3.2)–(3.4) can be re-organized as

$$\partial f_k(x_{k+1}) + A^{\mathrm{T}}\lambda_{k+1} + \rho_{k+1}A^{\mathrm{T}}B(z_k - z_{k+1}) + \eta_{k+1}(x_{k+1} - x_k) \ni 0, \quad (7.1)$$

$$B^{\mathrm{T}}\lambda_{k+1} = 0, \quad (7.2)$$

$$\lambda_{k+1} - \lambda_k - \rho_{k+1}(Ax_{k+1} + Bz_{k+1}) = 0. \quad (7.3)$$

Recalling $\lambda = [\beta; \gamma]$ and $B = [-I_{mp}; -I_{mp}]$, we have $\beta_{k+1} = -\gamma_{k+1}$ from (7.2). Utilizing this fact and the definitions $E_{\mathrm{o}} = A_{\mathrm{s}} - A_{\mathrm{d}}$, $E_{\mathrm{u}} = A_{\mathrm{s}} + A_{\mathrm{d}}$, $A = [A_{\mathrm{s}}; A_{\mathrm{d}}]$ and $B = [-I_{mp}; -I_{mp}]$, (7.1) becomes

$$\partial f_k(x_{k+1}) + E_{\mathrm{o}}^{\mathrm{T}}\beta_{k+1} - \rho_{k+1}E_{\mathrm{u}}^{\mathrm{T}}(z_k - z_{k+1}) + \eta_{k+1}(x_{k+1} - x_k) \ni 0. \quad (7.4)$$

If we initialize $\beta_1 = -\gamma_1$ such that $\beta_k = -\gamma_k$ holds for all times k, then (7.3) can be separated into two equations

$$\beta_{k+1} - \beta_k - \rho_{k+1}(A_{\mathrm{s}}x_{k+1} - z_{k+1}) = 0, \quad (7.5)$$

$$-\beta_{k+1} + \beta_k - \rho_{k+1}(A_{\mathrm{d}}x_{k+1} - z_{k+1}) = 0. \quad (7.6)$$

Subtracting (7.6) from (7.5) yields

$$\beta_{k+1} - \beta_k - \frac{\rho_{k+1}}{2}E_{\mathrm{o}}x_{k+1} = 0, \quad (7.7)$$

since $E_o = A_s - A_d$. On the other hand, summing up (7.6) and (7.5) yields $\frac{1}{2}E_u x_{k+1} - z_{k+1} = 0$ since $E_u = A_s + A_d$; we further initialize $\frac{1}{2}E_u x_1 = z_1$ such that for all times k, it holds

$$\frac{1}{2}E_u x_k - z_k = 0. \tag{7.8}$$

Substituting (7.8) into (7.4), we have

$$\partial f_k(x_{k+1}) + E_o^T \beta_{k+1} + \frac{\rho_{k+1}}{2} E_u^T E_u x_{k+1} - \frac{\rho_{k+1}}{2} E_u^T E_u x_k + \eta_{k+1}(x_{k+1} - x_k) \ni 0. \tag{7.9}$$

Defining a new multiplier $\alpha_k = E_o^T \beta_k$, from (7.7) and the definition $L_o = \frac{1}{2}E_o^T E_o$, we have (3.5). Replacing the term $E_o^T \beta_{k+1} = \alpha_{k+1}$ by $\alpha_k + \rho_{k+1} L_o x_{k+1}$ as shown in (3.5), as well as using the definition $L_u = \frac{1}{2}E_u^T E_u$ and the equality $2D = L_o + L_u$, we can simplify (7.9) as (3.6).

Appendix 2: Proof of Lemma 4.2

Proof Recall the definition of $\mathcal{R}_G^j$, we have

$$\begin{aligned}\mathcal{R}_G^j &= \sum_{k=1}^{T}\sum_{i=1}^{n}\left(f_k^i(x_k^j) - f_k^i(x^*)\right) \\ &= \sum_{k=1}^{T}\sum_{i=1}^{n}\left(f_k^i(x_k^i) - f_k^i(x^*)\right) + \sum_{k=1}^{T}\sum_{i=1}^{n}\left(f_k^i(x_k^j) - f_k^i(x_k^i)\right).\end{aligned} \tag{8.1}$$

In (8.1), the first summation is the definition of the nominal regret $\mathcal{R}_N$. The second summation, which describes the discrepancy of the local cost function values, is no greater than the discrepancy of the local estimates plus the upper bound of the subgradients. Considering that f_k^i is convex and using Young's inequality, since ρ_k, n, and σ are positive, it holds

$$\begin{aligned}f_k^i(x_k^j) - f_k^i(x_k^i) &\leqslant \langle \partial f_k^i(x_k^j), x_k^j - x_k^i \rangle \\ &\leqslant \frac{\rho_k}{2n\sigma}\left\|x_k^j - x_k^i\right\|^2 + \frac{n\sigma}{2\rho_k}\left\|\partial f_k^i(x_k^j)\right\|^2.\end{aligned} \tag{8.2}$$

Under Assumption 4.1 we know that $\|\partial f_k^i\| \leqslant L_f$; hence (8.2) becomes

$$f_k^i(x_k^j) - f_k^i(x_k^i) \leqslant \frac{\rho_k}{2n\sigma}\left\|x_k^j - x_k^i\right\|^2 + \frac{n\sigma L_f^2}{2\rho_k}. \tag{8.3}$$

Substituting (8.3) into (8.1),

$$\mathcal{R}_G^j \leqslant \mathcal{R}_N + \sum_{k=1}^{T}\frac{\rho_k}{2n\sigma}\sum_{i=1}^{n}\left\|x_k^j - x_k^i\right\|^2 + \frac{n^2\sigma L_f^2}{2}\sum_{k=1}^{T}\frac{1}{\rho_k}. \tag{8.4}$$

Further using the definition $\mathcal{R}_{\mathrm{G}} = \max_j \mathcal{R}_{\mathrm{G}}^j$ yields the upper bound of the global regret $\mathcal{R}_{\mathrm{G}}$ in (4.1).

Appendix 3: Proof of Lemma 4.3

Proof According to the definition of the vectors x_k and z_k as well as the matrices A and B

$$\|Ax_k + Bz_k\|^2 = \sum_{(i,j)\in\mathcal{A}} \left(\|x_k^i - z_k^{ij}\|^2 + \|x_k^j - z_k^{ij}\|^2\right). \tag{9.1}$$

Therefore, we can interpret $\|Ax_k + Bz_k\|^2$ by the summation of $\|x_k^i - x_k^j\|^2$ over all arcs (i, j)

$$\begin{aligned}\|Ax_k + Bz_k\|^2 &\geqslant \frac{1}{2}\sum_{(i,j)\in\mathcal{A}} \left\|\left(x_k^i - z_k^{ij}\right) - \left(x_k^j - z_k^{ij}\right)\right\|^2 \\ &= \frac{1}{2}\sum_{(i,j)\in\mathcal{A}} \left\|x_k^i - x_k^j\right\|^2. \end{aligned}\tag{9.2}$$

Consider a shortest path $i = v_0 \leftrightarrow v_1 \leftrightarrow \cdots \leftrightarrow v_{\sigma_{ij}-1} \leftrightarrow v_{\sigma_{ij}} = j$ between i and j; for any integer $\ell \in [0, \sigma_{ij} - 1]$, $(v_\ell, v_{\ell+1}) \in \mathcal{A}$. Notice that the network is bidirectionally connected such both (i, j) and (j, i) belong to $\mathcal{A}$, we have

$$\frac{1}{2}\sum_{(i,j)\in\mathcal{A}} \left\|x_k^i - x_k^j\right\|^2 \geqslant \left\|x_k^{v_0} - x_k^{v_1}\right\|^2 + \cdots + \left\|x_k^{v_{\sigma_{ij}-1}} - x_k^{v_{\sigma_{ij}}}\right\|^2. \tag{9.3}$$

Observe that the right-hand side of (9.3) has a lower bound

$$\begin{aligned}&\left\|x_k^{v_0} - x_k^{v_1}\right\|^2 + \cdots + \left\|x_k^{v_{\sigma_{ij}-1}} - x_k^{v_{\sigma_{ij}}}\right\|^2 \\ &\geqslant \frac{1}{\sigma_{ij}}\left\|\left(x_k^{v_0} - x_k^{v_1}\right) + \cdots + \left(x_k^{v_{\sigma_{ij}-1}} - x_k^{v_{\sigma_{ij}}}\right)\right\|^2 \\ &= \frac{1}{\sigma_{ij}}\|x_k^i - x_k^j\|^2. \end{aligned}\tag{9.4}$$

Combining (9.2), (9.3), and (9.4) completes the proof.

Appendix 4: Proof of Lemma 4.4

Proof From (7.2), for all times k, it holds $B^{\mathrm{T}}\lambda_{k+1} = 0$. With the initialization $\lambda_1 = 0$ for all times k, we know that $B^{\mathrm{T}}\lambda_k = 0$, which implies that

$$\langle \lambda_{k+1} - \lambda_k, Bz_k - Bz_{k+1}\rangle = 0. \tag{10.1}$$

Substituting the λ-update $\lambda_{k+1} - \lambda_k = \rho_{k+1}(Ax_{k+1} + Bz_{k+1})$ in (3.4) into (10.1) yields

$$\rho_{k+1}\langle Ax_{k+1} + Bz_{k+1}, Bz_k - Bz_{k+1}\rangle = 0, \tag{10.2}$$

or equivalently

$$\frac{\rho_{k+1}}{2}\left\{\|Ax_{k+1} + Bz_k\|^2 - \|Ax_{k+1} + Bz_{k+1}\|^2 - \|Bz_k - Bz_{k+1}\|^2\right\} = 0. \tag{10.3}$$

Since $\rho_{k+1} > 0$, (10.3) implies that $\|Ax_{k+1} + Bz_k\|^2 - \|Ax_{k+1} + Bz_{k+1}\|^2 \geqslant 0$, which completes the proof.

Appendix 5: Proof of Lemma 4.5

Proof From (4.1) in Lemma 4.2 under the initialization $x_1 = 0$

$$\begin{aligned}\mathcal{R}_{\mathrm{G}} &\leqslant \mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T}\frac{\rho_k}{2n\sigma}\max_j\sum_{i=1}^{n}\left\|x_k^j - x_k^i\right\|^2 + \frac{n^2\sigma L_f^2}{2}\sum_{k=1}^{T}\frac{1}{\rho_k}\\ &\leqslant \mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T}\frac{\rho_{k+1}}{2n\sigma}\max_j\sum_{i=1}^{n}\left\|x_{k+1}^j - x_{k+1}^i\right\|^2 + \frac{n^2\sigma L_f^2}{2}\sum_{k=1}^{T}\frac{1}{\rho_k}.\end{aligned} \tag{11.1}$$

Combining the two inequalities, (4.2) in Lemma 4.3 and (4.3) in Lemma 4.4, $\|x_{k+1}^i - x_{k+1}^j\|^2 \leqslant \sigma_{ij}\|Ax_{k+1} + Bz_{k+1}\|^2 \leqslant \sigma_{ij}\|Ax_{k+1} + Bz_k\|^2$. Therefore, (11.1) boils down to

$$\mathcal{R}_{\mathrm{G}} \leqslant \mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T}\frac{\rho_{k+1}}{2n\sigma}\max_j\sum_{i=1}^{n}\sigma_{ij}\|Ax_{k+1} + Bz_k\|^2 + \frac{n^2\sigma L_f^2}{2}\sum_{k=1}^{T}\frac{1}{\rho_k}. \tag{11.2}$$

According to the definition of the network diameter $\sigma = \max_{i,j}\sigma_{ij}$, we have (4.4).

Appendix 6: Proof of Theorem 4.6

Proof According to Lemma 4.5, the global regret $\mathcal{R}_{\mathrm{G}}$ is no greater than the summation of three components: the nominal regret $\mathcal{R}_{\mathrm{N}}$, the accumulated constraint violation, and the accumulated inverse of stepsize. In the proof, we first establish an $O(\sqrt{T})$ bound for the summation of the first two components, and then establish an $O(\sqrt{T})$ bound for the last component.

Recalling that $f_k(x) := \sum_{i=1}^n f_k^i(x^i)$, $x_k = [x_k^1; x_k^2; \cdots; x_k^n]$, and $x^* = [\tilde{x}^*; \tilde{x}^*; \cdots; \tilde{x}^*]$, we can rewrite the instantaneous nominal regret at time k as

$$\begin{aligned}\sum_{i=1}^{n}\left(f_k^i(x_k^i) - f_k^i(\tilde{x}^*)\right) &= f_k(x_k) - f_k(x^*)\\ &= \left(f_k(x_k) - f_k(x_{k+1})\right) + \left(f_k(x_{k+1}) - f_k(x^*)\right).\end{aligned} \tag{12.1}$$

Below we consider $f_k(x_k) - f_k(x_{k+1})$ and $f_k(x_{k+1}) - f_k(x^*)$ respectively.

Since the local cost functions f_k^i are convex, the aggregated cost function f_k is also convex. Therefore, it holds

$$f_k(x_k) - f_k(x_{k+1}) \leqslant \langle \partial f_k(x_k), x_k - x_{k+1} \rangle. \tag{12.2}$$

Applying Young's inequality to the right-hand side of (12.2), for any finite and positive η_{k+1}, we have

$$f_k(x_k) - f_k(x_{k+1}) \leqslant \frac{1}{2\eta_{k+1}} \|\partial f_k(x_k)\|^2 + \frac{\eta_{k+1}}{2} \|x_k - x_{k+1}\|^2. \tag{12.3}$$

On the other hand, the convexity of f_k implies that

$$f_k(x_{k+1}) - f_k(x^*) \leqslant \langle \partial f_k(x_{k+1}), x_{k+1} - x^* \rangle. \tag{12.4}$$

Substituting the expression of $f_k(x_{k+1})$ in (7.1) into (12.4) and utilizing the equality $Ax^* + Bz^* = 0$ yield

$$\begin{aligned} f_k(x_{k+1}) - f_k(x^*) &\leqslant \langle -A^{\mathrm{T}}\lambda_{k+1} - \rho_{k+1}A^{\mathrm{T}}(Bz_k - Bz_{k+1}) \\ &\quad - \eta_{k+1}(x_{k+1} - x_k), x_{k+1} - x^* \rangle \\ &= -\langle \lambda_{k+1}, Ax_{k+1} - Ax^* \rangle - \rho_{k+1}\langle Bz_k - Bz_{k+1}, Ax_{k+1} - Ax^* \rangle \\ &\quad - \eta_{k+1}\langle x_{k+1} - x_k, x_{k+1} - x^* \rangle \\ &= -\langle \lambda_{k+1}, Ax_{k+1} + Bz^* \rangle - \rho_{k+1}\langle Bz_k - Bz_{k+1}, Ax_{k+1} + Bz^* \rangle \\ &\quad - \eta_{k+1}\langle x_{k+1} - x_k, x_{k+1} - x^* \rangle. \end{aligned} \tag{12.5}$$

Now reorganize the three terms in the right-hand side of (12.5). For the first term observing that $B^{\mathrm{T}}\lambda_{k+1} = 0$ as shown in (7.2), we can replace z^* by z_{k+1} that does not change the value. Further using the equality $\lambda_{k+1} - \lambda_k - \rho_{k+1}(Ax_{k+1} + Bz_{k+1}) = 0$ as shown in (7.3), we have

$$\begin{aligned} -\langle \lambda_{k+1}, Ax_{k+1} + Bz^* \rangle &= -\langle \lambda_{k+1}, Ax_{k+1} + Bz_{k+1} \rangle \\ &= \frac{1}{2\rho_{k+1}} \left\{ \|\lambda_k\|^2 - \|\lambda_{k+1}\|^2 \right\} - \frac{\rho_{k+1}}{2} \|Ax_{k+1} + Bz_{k+1}\|^2. \end{aligned} \tag{12.6}$$

For the second term, we have

$$\begin{aligned} &-\rho_{k+1}\langle Bz_k - Bz_{k+1}, Ax_{k+1} + Bz^* \rangle \\ &= \frac{\rho_{k+1}}{2} \left\{ \|Bz^* - Bz_k\|^2 - \|Bz^* - Bz_{k+1}\|^2 - \|Ax_{k+1} + Bz_k\|^2 \right. \\ &\quad \left. + \|Ax_{k+1} + Bz_{k+1}\|^2 \right\}. \end{aligned} \tag{12.7}$$

For the third term, we have

$$-\eta_{k+1}\langle x_{k+1}-x_k, x_{k+1}-x^*\rangle = \frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2 - \|x^*-x_{k+1}\|^2 - \|x_{k+1}-x_k\|^2\right\}. \tag{12.8}$$

Substituting (12.6)–(12.8) into (12.5) yields

$$\begin{aligned} & f_k(x_{k+1}) - f_k(x^*) \\ \leqslant & \frac{1}{2\rho_{k+1}}\left\{\|\lambda_k\|^2 - \|\lambda_{k+1}\|^2\right\} + \frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2 - \|Bz^*-Bz_{k+1}\|^2\right. \\ & \left. - \|Ax_{k+1}+Bz_k\|^2\right\} + \frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2 - \|x^*-x_{k+1}\|^2 - \|x_{k+1}-x_k\|^2\right\}. \end{aligned} \tag{12.9}$$

Combining (12.3) and (12.9), we obtain an upper bound for $f_k(x_k) - f_k(x^*)$ that is

$$\begin{aligned} & f_k(x_k) - f_k(x^*) \\ \leqslant & \frac{1}{2\rho_{k+1}}\left\{\|\lambda_k\|^2 - \|\lambda_{k+1}\|^2\right\} + \frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2 - \|Bz^*-Bz_{k+1}\|^2\right. \\ & \left. - \|Ax_{k+1}+Bz_k\|^2\right\} + \frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2 - \|x^*-x_{k+1}\|^2\right\} + \frac{1}{2\eta_{k+1}}\|\partial f_k(x_k)\|^2, \end{aligned} \tag{12.10}$$

which immediately yields

$$\begin{aligned} & f_k(x_k) - f_k(x^*) + \frac{\rho_{k+1}}{2}\|Ax_{k+1}+Bz_k\|^2 \\ \leqslant & \frac{1}{2\rho_{k+1}}\left\{\|\lambda_k\|^2 - \|\lambda_{k+1}\|^2\right\} + \frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2 - \|Bz^*-Bz_{k+1}\|^2\right\} \\ & + \frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2 - \|x^*-x_{k+1}\|^2\right\} + \frac{1}{2\eta_{k+1}}\|\partial f_k(x_k)\|^2. \end{aligned} \tag{12.11}$$

Notice that summing up the left-hand side of (12.11) from $k=1$ to $k=T$ leads to the nominal regret $\mathcal{R}_{\mathrm{N}}$ plus the accumulated constraint violation; see the definition of $\mathcal{R}_{\mathrm{N}}$ and (4.4) in Lemma 4.5. Since ρ_{k+1} and η_{k+1} are both constants here, we have

$$\begin{aligned} & \mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\|Ax_{k+1}+Bz_k\|^2 \\ \leqslant & \frac{1}{2\rho_{k+1}}\left\{\|\lambda_1\|^2 - \|\lambda_{T+1}\|^2\right\} + \frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_1\|^2 - \|Bz^*-Bz_{T+1}\|^2\right\} \\ & + \frac{\eta_{k+1}}{2}\left\{\|x^*-x_1\|^2 - \|x^*-x_{T+1}\|^2\right\} + \frac{1}{2\eta_{k+1}}\sum_{k=1}^{T}\|\partial f_k(x_k)\|^2 \end{aligned}$$

$$\leqslant \frac{1}{2\rho_{k+1}}\|\lambda_1\|^2+\frac{\rho_{k+1}}{2}\|Bz^*-Bz_1\|^2+\frac{\eta_{k+1}}{2}\|x^*-x_1\|^2$$
$$+\frac{1}{2\eta_{k+1}}\sum_{k=1}^{T}\|\partial f_k(x_k)\|^2. \tag{12.12}$$

Observing that $\|x^*\| \leqslant \sqrt{n}D_{\tilde{x}}$ and $\|Bz^*\| \leqslant \sqrt{2m}D_{\tilde{x}}$, $||\partial f_k|| \leqslant \sqrt{n}L_f$, $\rho_{k+1} = c_1 n\sqrt{T}$, $\eta_{k+1} = (L_f/D_{\tilde{x}})\sqrt{T}$, and the algorithm is initialized by $x_1 = 0$, $z_1 = 0$, and $\lambda_1 = 0$, (12.12) boils down to

$$\mathcal{R}_{\mathrm{N}}+\sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\|Ax_{k+1}+Bz_k\|^2 \leqslant \rho_{k+1}mD_{\tilde{x}}^2+\eta_{k+1}\frac{nD_{\tilde{x}}^2}{2}+\frac{1}{\eta_{k+1}}\frac{nL_f^2T}{2}$$
$$=\left(c_1 nmD_{\tilde{x}}^2+nL_fD_{\tilde{x}}\right)\sqrt{T}. \tag{12.13}$$

Second we consider the accumulated inverse of stepsize. Since $\rho_k = c_1 n\sqrt{T}$, we have

$$\frac{n^2\sigma L_f^2}{2}\sum_{k=1}^{T}\frac{1}{\rho_k}=\frac{n\sigma L_f^2}{2c_1}\sqrt{T}. \tag{12.14}$$

Substituting (12.13) and (12.14) into (4.4), we obtain the $O(\sqrt{T})$ bound of the global regret $\mathcal{R}_{\mathrm{G}}$ in (4.5).

Appendix 7: Proof of Theorem 4.7

Proof Similar to the proof of Theorem 4.6, we first establish an $O(\log T)$ bound for the summation of the nominal regret $\mathcal{R}_{\mathrm{N}}$ and the accumulated constraint violation, and then establish an $O(\log T)$ bound for the accumulated inverse of stepsize.

Consider the summation of the nominal regret $\mathcal{R}_{\mathrm{N}}$ and the accumulated constraint violation. Since the local cost functions f_k^i are μ-strongly convex, the aggregated cost function f_k is also μ-strongly convex. Therefore, we have

$$f_k(x_{k+1})-f_k(x^*) \leqslant \langle \partial f_k(x_{k+1}), x_{k+1}-x^*\rangle-\frac{\mu}{2}\|x^*-x_{k+1}\|^2, \tag{13.1}$$

which modifies (12.4) by subtracting a quadratic term $\frac{\mu}{2}\|x^*-x_{k+1}\|^2$ at the right-hand side. Similar to the proof of Theorem 4.6 through (12.1) to (12.11), we have

$$f_k(x_k)-f_k(x^*)+\frac{\rho_{k+1}}{2}\|Ax_{k+1}+Bz_k\|^2$$
$$\leqslant \frac{1}{2\rho_{k+1}}\left\{\|\lambda_k\|^2-\|\lambda_{k+1}\|^2\right\}+\frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2-\|Bz^*-Bz_{k+1}\|^2\right\}$$

$$+\frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2-\|x^*-x_{k+1}\|^2\right\}-\frac{\mu}{2}\|x^*-x_{k+1}\|^2$$
$$+\frac{1}{2\eta_{k+1}}\|\partial f_k(x_k)\|^2. \tag{13.2}$$

Therefore, summing up from $k=1$ to $k=T$ yields

$$\begin{aligned}
&\mathcal{R}_{\mathrm{N}}+\sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\|Ax_{k+1}+Bz_k\|^2\\
&\leqslant\sum_{k=1}^{T}\frac{1}{2\rho_{k+1}}\left\{\|\lambda_k\|^2-\|\lambda_{k+1}\|^2\right\}\\
&\quad+\sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2-\|Bz^*-Bz_{k+1}\|^2\right\}\\
&\quad+\sum_{k=1}^{T}\frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2-\|x^*-x_{k+1}\|^2\right\}-\sum_{k=1}^{T}\frac{\mu}{2}\|x^*-x_{k+1}\|^2\\
&\quad+\sum_{k=1}^{T}\frac{1}{2\eta_{k+1}}\|\partial f_k(x_k)\|^2.
\end{aligned}\tag{13.3}$$

Below we find upper bounds for the right-hand side terms in (13.3). First observe that ρ_k is non-decreasing and $\lambda_1=0$ by initialization

$$\sum_{k=1}^{T}\frac{1}{2\rho_{k+1}}\left\{\|\lambda_k\|^2-\|\lambda_{k+1}\|^2\right\}\leqslant\frac{1}{2\rho_2}\|\lambda_1\|^2=0. \tag{13.4}$$

Second observe that $B=[-I_{mp};-I_{mp}]$, we have

$$\|z^*-z_{k+1}\|^2=\frac{1}{2}\|Bz^*-Bz_{k+1}\|^2. \tag{13.5}$$

Then for any constant $c_2>0$, it holds

$$\begin{aligned}
&\sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2-\|Bz^*-Bz_{k+1}\|^2\right\}-\sum_{k=1}^{T}c_2\|z^*-z_{k+1}\|^2\\
&=\sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\left\{\|Bz^*-Bz_k\|^2-\|Bz^*-Bz_{k+1}\|^2\right\}-\sum_{k=1}^{T}\frac{c_2}{2}\|Bz^*-Bz_{k+1}\|^2\\
&=\frac{\rho_2}{2}\|Bz^*-Bz_1\|^2+\sum_{k=2}^{T}\frac{\rho_{k+1}-\rho_k-c_2}{2}\|Bz^*-Bz_k\|^2.
\end{aligned}\tag{13.6}$$

Third observe that $\frac{1}{2}E_{\mathrm{u}}x_k - z_k = 0$ in (7.8) and $\frac{1}{2}E_{\mathrm{u}}x^* - z^* = 0$ by definition. Using the fact that the maximum singular value of E_{u} is $\sqrt{2\Gamma_{\mathrm{u}}}$, we have

$$\|z^* - z_{k+1}\|^2 = \|\frac{1}{2}E_{\mathrm{u}}(x^* - x_{k+1})\|^2 \leqslant \frac{\Gamma_{\mathrm{u}}}{2}\|x^* - x_{k+1}\|^2. \tag{13.7}$$

Then for any constant $c_2 > 0$, it holds

$$\begin{aligned}
&\sum_{k=1}^{T}\frac{\eta_{k+1}}{2}\left\{\|x^*-x_k\|^2-\|x^*-x_{k+1}\|^2\right\}-\sum_{k=1}^{T}\frac{\mu}{2}\|x^* - x_{k+1}\|^2\\
&+\sum_{k=1}^{T}c_2\|z^* - z_{k+1}\|^2\\
&\leqslant \sum_{k=1}^{T}\frac{\eta_{k+1}}{2}\left\{\|x^* - x_k\|^2 - \|x^* - x_{k+1}\|^2\right\} - \sum_{k=1}^{T}\frac{\mu - c_2\Gamma_{\mathrm{u}}}{2}\|x^* - x_{k+1}\|^2\\
&= \frac{\eta_2}{2}\|x^* - x_1\|^2 + \sum_{k=2}^{T}\frac{\eta_{k+1} - \eta_k - \mu + c_2\Gamma_{\mathrm{u}}}{2}\|x^* - x_k\|^2.
\end{aligned} \tag{13.8}$$

Summing up (13.4), (13.6), and (13.8) and substituting into the right-hand side of (13.3), we have

$$\begin{aligned}
&\mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T}\frac{\rho_{k+1}}{2}\|Ax_{k+1} + Bz_k\|^2\\
&\leqslant \frac{\rho_2}{2}\|Bz^* - Bz_1\|^2 + \sum_{k=2}^{T}\frac{\rho_{k+1} - \rho_k - c_2}{2}\|Bz^* - Bz_k\|^2\\
&\quad + \frac{\eta_2}{2}\|x^* - x_1\|^2 + \sum_{k=2}^{T}\frac{\eta_{k+1} - \eta_k - \mu + c_2\Gamma_{\mathrm{u}}}{2}\|x^* - x_k\|^2\\
&\quad + \sum_{k=1}^{T}\frac{1}{2\eta_{k+1}}\|\partial f_k(x_k)\|^2,
\end{aligned} \tag{13.9}$$

which holds for any constant $c_2 > 0$. Letting $\rho_k = c_2 k$ and $\eta_k = (\mu - c_2\Gamma_{\mathrm{u}})k$, we are able to eliminate the summations over $\|Bz^* - Bz_k\|^2$ and $\|x^* - x_k\|^2$.

Specifically we set $c_2 = \frac{\mu}{2\Gamma_{\mathrm{u}}}$ such that $\rho_k = \frac{\mu}{2\Gamma_{\mathrm{u}}}k$ and $\eta_k = \frac{\mu}{2}k$. Since $z_1 = 0$ and $x_1 = 0$ by definition, $\|x^*\| \leqslant D_x$, $\|Bz^*\| \leqslant \frac{\sqrt{2m}D_x}{\sqrt{n}}$, and $||\partial f_k|| \leqslant \sqrt{n}L_{\mathrm{f}}$ by hypothesis, we obtain an upper bound for the right-hand side of (13.9) as

$$\mathcal{R}_{\mathrm{N}} + \sum_{k=1}^{T} \frac{\rho_{k+1}}{2} \|Ax_{k+1} + Bz_k\|^2 \leqslant \frac{\mu m}{\Gamma_{\mathrm{u}} n} D_x^2 + \frac{\mu}{2} D_x^2 + \sum_{k=1}^{T} \frac{n L_f^2}{\mu(k+1)}$$
$$\leqslant \frac{2m + \Gamma_{\mathrm{u}} n}{2\Gamma_{\mathrm{u}} n} \mu D_x^2 + \frac{n L_f^2}{\mu} (\log T + 1). \quad (13.10)$$

To obtain the inequality, we use the fact $\sum_{k=1}^{T} \frac{1}{k+1} \leqslant \int_{k=0}^{T} \frac{1}{k+1} \mathrm{d}k = \log(T+1) \leqslant \log T + 1$.

Consider the accumulated inverse of stepsize. Since $\rho_k = \frac{\mu}{2\Gamma_{\mathrm{u}}} k$, we have

$$\frac{n^2 \sigma L_f^2}{2} \sum_{k=1}^{T} \frac{1}{\rho_k} = \frac{\Gamma_{\mathrm{u}} n^2 \sigma L_f^2}{\mu} \sum_{k=1}^{T} \frac{1}{k} \leqslant \frac{\Gamma_{\mathrm{u}} n^2 \sigma L_f^2}{\mu} (\log T + 1). \quad (13.11)$$

To obtain the inequality we use the fact $\sum_{k=1}^{T} \frac{1}{k} = \sum_{k=1}^{T-1} \frac{1}{k+1} + 1 \leqslant \int_{k=0}^{T-1} \frac{1}{k+1} \mathrm{d}k + 1 = \log T + 1$.

Substituting (13.10) and (13.11) into (4.4), we obtain the $O(\log T)$ bound of the global regret $\mathcal{R}_{\mathrm{G}}$ in (4.6).

References

[1] Shalev-Shwartz, S.: Online learning and online convex optimization. Found. Trends Mach. Learn. **4**, 107–194 (2011)

[2] Schizas, I., Ribeiro, A., Giannakis, G.: Consensus in ad hoc WSNs with noisy links—Part I: distributed estimation of deterministic signals. IEEE Trans. Signal Process. **56**, 350–364 (2008)

[3] Nedic, A., Ozdaglar, A.: Distributed subgradient methods for multi-agent optimization. IEEE Trans. Autom. Control **54**, 48–61 (2009)

[4] Duchi, J., Agarwal, A., Wainwright, M.: Dual averaging for distributed optimization: convergence analysis and network scaling. IEEE Trans. Autom. Control **57**, 592–606 (2012)

[5] Shi, W., Ling, Q., Wu, G., Yin, W.: EXTRA: an exact first-order algorithm for decentralized consensus optimization. SIAM J. Optim. **25**, 944–966 (2015)

[6] Yuan, K., Ling, Q., Yin, W.: On the convergence of decentralized gradient descent. SIAM J. Optim. http://www.optimization-online.org/DB_FILE/2013/10/4097.pdf

[7] Yan, F., Sundaram, S., Vishwanathan, S., Qi, Y.: Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties. IEEE Trans. Knowl. Data Eng. **25**, 2483–2493 (2013)

[8] Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of International Conference on Machine Learning (2003)

[9] Hazan, E., Agarwal, A., Kale, S.: Logarithmic regret algorithms for online convex optimization. Mach. Learn. **69**, 169–192 (2007)

[10] Hosseini, S., Chapman, A., Mesbahi, M.: Online distributed optimization via dual avaraging. In: Proceedings of IEEE Conference on Decision and Control (2013)

[11] Tsianos, K., Rabbat, M.: Distributed strongly convex optimization. In: Proceedings of Allerton Conference on Communication, Control, and Computing (2012)

[12] Xiao, L.: Dual averaging methods for regularized stochastic learning and online optimization. J. Mach. Learn. Res. **11**, 2543–2596 (2010)

[13] Koppel, A., Jakubiec, F., Ribeiro, A.: A saddle point algorithm for networked online convex optimization, In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (2014)

[14] Nedic, A., Ozdaglar, A.: Subgradient methods for saddle-point problems. J. Optim. Theory Appl. **142**, 205–228 (2009)

[15] Mateos, G., Bazerque, J., Giannakis, G.: Distributed sparse linear regression. IEEE Trans. Signal Process. **58**, 5262–5276 (2010)

[16] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2010)
[17] Bazerque, J., Mateos, G., Giannakis, G.: Group-lasso on splines for spectrum cartograph. IEEE Trans. Signal Process. **59**, 4648–4663 (2011)
[18] Kekatos, V., Giannakis, G.: Distributed robust power system state estimation. IEEE Trans. Power Syst. **28**, 1617–1626 (2013)
[19] Eckstein, J., Bertsekas, D.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**, 293–318 (1992)
[20] Bertsekas, D., Tsitsiklis, J.: Parallel and Distributed Computation: Numerical Methods, 2nd edn. Athena Scientific, Belmont (1997)
[21] Shi, W., Ling, Q., Yuan, K., Wu, G., Yin, W.: On the linear convergence of the ADMM in decentralized consensus optimization. IEEE Trans. Signal Process. **62**, 1750–1761 (2014)
[22] Ling, Q., Ribeiro, A.: Decentralized dynamic optimization through the alternating direction method of multipliers. IEEE Trans. Signal Process. **62**, 1185–1197 (2014)
[23] Wang, H., Banerjee, A.: Online alternating direction method, In: Proceedings of International Conference on Machine Learning (2012)
[24] Libsvm dataset available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
[25] Vapnik, V.: Statistical Learning Theory. Wiley-Interscience, New York (1998)