文章编号 1672-6634(2023)05-0001-12 DOI 10.19728/j.issn1672－6634.2023040011

在线分布式优化研究进展

袁德明[1],张保勇[1],夏建伟[2]

(1. 南京理工大学 自动化学院,江苏 南京 210094;2. 聊城大学 数学科学学院, 山东 聊城 252059)

摘 要 在线分布式优化是通过多智能体之间有效的协调合作以实现实时优化的任务,其在多无人系统实时编队跟踪、大规模分布式机器学习、传感器网络的动态参数估计等领域中有着广泛的应用。分别从决策空间、性能指标、信息反馈模型等角度对近年来国内外在线分布式优化的代表性研究工作进行了梳理和总结,着重从算法设计思路和收敛性结果两个角度进行了剖析,同时也指出不同算法的优势和不足。最后,对在线分布式优化算法的应用进行了简单讨论并对全文进行了总结和展望。

关键词 在线分布式优化;多智能体系统;遗憾;信息反馈模型

中图分类号 TP13

文献标识码 A 开放科学(资源服务)标识码(OSID)

Recent Advance in Online Distributed Optimization

YUAN Deming[1], ZHANG Baoyong[1], XIA Jianwei[2]

(1. School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China;
2. School of Mathematical Sciences, Liaocheng University, Liaocheng 252059, China)

Abstract Online distributed optimization is a real-time optimization task fulfilled by a system of multiple agents through effective cooperation, and it can be applied to fields such as formation and tracking of multiple unmanned system, large-scale distributed machine learning, and dynamic parameter estimation in sensor networks. This paper presents an overview of the state-of-the-art research on online distributed optimization in recent years, from the angle of decision space, performance index, information feedback model, with special focus on the intuition behind the algorithm design and the convergence analysis results. Meanwhile, discussions about the pros and cons of the algorithms have been provided. Finally, we present some applications of online distributed optimization, make a summary of the paper, and discuss possible directions for future research.

Key words online distributed optimization; multi-agent systems; regret; information feedback model

收稿日期:2023-04-27

基金项目:国家自然科学基金(62022042, 62273181,61973148);中央高校基本科研业务费专项资金(30919011105)资助

通讯作者:袁德明,男,汉族,工学博士,教授,博士生导师,国家自然科学基金优秀青年基金获得者,研究方向:分布式优化、控制与机器学习,E-mail:dmyuan1012@njust.edu.cn。

1 引言

近年来,多智能体系统技术作为改变未来社会生活方式和战争规则的颠覆性技术在世界范围内得到快速发展,不仅成为当前国际学术界和产业界的研究热点,而且已经上升为我国国家战略的核心内容。多智能体系统(Multi-Agent Systems)是由一群具备一定的感知、通信、计算和执行能力的智能体通过通讯等方式关联成的一个网络系统[1,2]。而分布式优化是通过多智能体之间的协调合作有效地实现优化的任务。与传统集中式优化相比,分布式协同优化是一种"去中心"的优化算法,不需要一对一的全局通信,每个智能体只需与其相邻智能体进行信息交换。因此,可以用来解决许多传统集中式算法难以胜任的大规模复杂的优化问题。近十年来,因在机器学习、智能电网、传感器网络等诸多领域的广泛应用,分布式优化研究得到了越来越多国内外学者的广泛关注,是当今系统控制重要前沿发展方向之一[3-12]。

分布式优化问题在优化领域由来已久。早在 20 世纪 80 年代,麻省理工学院 John N. Tsitsiklis,Dimitri P. Bertsekas 等教授在文献[3]中提出了分布式优化理论的计算框架。基于此开创性工作,Angelia Nedic 教授和 Asuman Ozdaglar 教授首次在多智能体系统一致性框架下研究分布式优化问题,并提出了著名的分布式次梯度算法[4]。此算法由两部分构成:局部优化和一致性计算,前者是每个智能体使用次梯度算法来优化其自身的目标函数,而后者是用一致性算法来协调每个智能体和其相邻智能体的信息,以保持状态一致。基于此工作,近年来国内外学者在分布式优化上取得了大量重要的研究成果,并见刊于系统控制领域国内外权威刊物和会议上[5-12]。

在线分布式优化的研究近年来得到了国内外学者的广泛关注。与离线分布式优化不同的是,在线分布式优化具有如下特征:(1) 优化目标不同:在线分布式优化中,智能体的目标函数是时变的,其产生甚至可以基于智能体历史的决策信息和损失函数信息;而离线分布式优化中,智能体的目标函数是固定不变的;(2) 性能指标不同:在分析在线分布式优化算法的性能时,通常将网络中的任意一个智能体的决策序列在所有损失函数上产生的损失与最优决策所产生的损失相比较,其差值称为遗憾(Regret);而离线分布式优化是最小化智能体的状态序列与最优值之差。因此,在线分布式优化是离线分布式优化的延伸和拓展。在线分布式优化根据决策空间类型可以分为一般凸集约束和不等式约束;根据信息反馈模型的不同可以分为完全信息下在线分布式优化和 Bandit 信息下在线分布式优化。本文将从决策空间、信息反馈模型这两个角度对在线分布式优化算法的最新研究成果进行总结,着重讨论算法的设计思路、参数设计以及收敛性分析结果,并对不同算法的优势和不足进行剖析。

本文结构安排如下:第 2 节介绍了在线分布式优化问题的模型和性能指标;第 3 节梳理了现有在线分布式优化相关的代表性算法及其收敛性结果;第 4 节简单讨论了在线分布式优化算法的应用;第 5 节对全文进行总结并浅析未来可能的研究方向。

下面介绍本文的记号。R ,R_+ ,R^d 分别表示实数空间,正实数空间和 d 维欧氏空间。对向量 $\boldsymbol{x}\in R^d$,$[\boldsymbol{x}]_i$ 表示向量 $\boldsymbol{x}$ 的第 i 个元素,$\boldsymbol{x}^{\mathrm{T}}$ 表示其转置,$\|\boldsymbol{x}\|_1$,$\|\boldsymbol{x}\|_2$,$\|\boldsymbol{x}\|_\infty$,$\|\boldsymbol{x}\|_p(p\geqslant 1)$ 分别表示其 1-范数,2-范数,无穷范数和 p -范数。$[\boldsymbol{P}]_{ij}$ 表示矩阵 $\boldsymbol{P}$ 第 i 行第 j 列的元素。记 $B(\boldsymbol{a},r)=\{\boldsymbol{x}\in R^d:\|\boldsymbol{x}-\boldsymbol{a}\|\leqslant r\}$。$\mathrm{Proj}_X[\cdot]$为欧几里得投影,其定义为 $\mathrm{Proj}_X[\boldsymbol{z}]=\arg\min_{\boldsymbol{x}\in X}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2$。为了方便讨论简要介绍凸集和凸函数的定义。令集合 $X\subseteq R^d$,如果对于任意的 $\boldsymbol{x},\boldsymbol{y}\in X$ 和 $0\leqslant\theta\leqslant 1$ 有 $\theta\boldsymbol{x}+(1-\theta)\boldsymbol{y}\in X$,则称 X 为凸集。令 $f(\boldsymbol{x}):X\to R$,如果对于任意的 $\boldsymbol{x},\boldsymbol{y}\in X$ 和 $0\leqslant\theta\leqslant 1$ 有 $f(\theta\boldsymbol{x}+(1-\theta)\boldsymbol{y})\leqslant\theta f(\boldsymbol{x})+(1-\theta)f(\boldsymbol{y})$,则称 $f(\boldsymbol{x})$ 为凸函数。对于凸函数 $f(\boldsymbol{x})$,下列一阶条件成立:对于任意的 $\boldsymbol{x},\bar{\boldsymbol{x}}\in X$,$f(\boldsymbol{x})\geqslant f(\bar{\boldsymbol{x}})+g(\bar{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{x}-\bar{\boldsymbol{x}})$,其中 $g(\bar{\boldsymbol{x}})\in\partial f(\bar{\boldsymbol{x}})$,而 $\partial f(\bar{\boldsymbol{x}})$ 为 $f(\boldsymbol{x})$ 在 $\bar{\boldsymbol{x}}$ 点的所有次梯度的集合(当 $f(\boldsymbol{x})$ 可微时,$\partial f(\bar{\boldsymbol{x}})=\{\nabla f(\bar{\boldsymbol{x}})\}$)。如果对于任意的 $\boldsymbol{x},\bar{\boldsymbol{x}}\in X$ 存在常数 $\sigma>0$ 使得 $f(\boldsymbol{x})\geqslant f(\boldsymbol{x}-)+g(\boldsymbol{x}-)^{\mathrm{T}}(\boldsymbol{x}\boldsymbol{x}-)+\frac{\sigma}{2}\|\boldsymbol{x}-\boldsymbol{x}-\|_2^2$,则称 $f(\boldsymbol{x})$ 是强凸函数。如果凸函数 $f(\boldsymbol{x})$ 具有 Lipschitz 连续梯度,即对于任意 $\boldsymbol{x},\bar{\boldsymbol{x}}\in$

X 存在常数 $L>0$ 使得 $\|\nabla f(\boldsymbol{x})-\nabla f(\bar{\boldsymbol{x}})\|_2 \leqslant L\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2$，那么 $f(\boldsymbol{x}) \leqslant f(\bar{\boldsymbol{x}})+g(\bar{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{x}-\bar{\boldsymbol{x}})+\frac{L}{2}\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2^2$。

2　模型与性能指标

2.1　模型

在线分布式优化以多智能体系统为框架。多智能体系统是一群具有一定感知、计算和通信能力的个体组成的网络系统。一般地，多智能体系统在 t 时刻的拓扑结构或信息分享模型可以由有向图 $G_t=(V,E_t,\boldsymbol{P}_t)$ 表示，其中 $V=\{1,2,\cdots,N\}$ 为顶点（智能体）集合，$E_t \subset V\times V$ 为 t 时刻的边集，$\boldsymbol{P}_t=([\boldsymbol{P}_t]_{ij})_{N\times N}$ 为 t 时刻的权重矩阵，它刻画了 t 时刻智能体之间的信息分享关系。下面介绍两种经典的网络拓扑，它们广泛地应用于在线分布式优化算法的设计和研究中。

（1）无向连通固定拓扑[7,12]：$G=(V,E,\boldsymbol{P})$ 为无向图，其中 E 为固定边集，$\boldsymbol{P}=([\boldsymbol{P}]_{ij})_{N\times N}$ 为固定的权重矩阵。无向图意味着对于任意的两个智能体 $i,j\in V$，$(i,j)\in E$ 当且仅当 $(j,i)\in E$；而图是连通的意味着对于任意的两个智能体 i 和 j 之间都存在一条路径。一般对权重矩阵做如下假设

(i) 对任意的 $i,j\in V$，$[\boldsymbol{P}]_{ij}\geqslant 0$；

(ii) 当 $(i,j)\in E$ 有 $[\boldsymbol{P}]_{ij}>0$；当 $(i,j)\notin E$ 有 $[\boldsymbol{P}]_{ij}=0$；

(iii) $\boldsymbol{P}$ 为双随机矩阵，即 $\boldsymbol{P}\mathbf{1}_N=\mathbf{1}_N$，$\mathbf{1}_N^{\mathrm{T}}\boldsymbol{P}=\mathbf{1}_N^{\mathrm{T}}$，其中 $\mathbf{1}_N=(1,\cdots,1)^{\mathrm{T}}\in R^N$。

在此假设下，权重矩阵 $\boldsymbol{P}$ 的第二大奇异值小于 1，即 $\sigma_2(\boldsymbol{P})<1$，它决定了一致性收敛速度的快慢。

（2）一致联合强连通切换拓扑[4,9]：存在一个固定的正整数 B 使得对于所有的 $k\geqslant 0$ 有 $(V,E_{kB}\cup E_{kB+1}\cup\cdots\cup E_{(k+1)B-1})$ 是强连通的，即图中的任意两个顶点之间存在有向路径。针对此拓扑，一般对权重矩阵做如下假设

(i) 对任意的 $i\in V$ 有 $[\boldsymbol{P}_t]_{ii}\geqslant\beta>0$；

(ii) 当 $(i,j)\in E_t$ 有 $[\boldsymbol{P}_t]_{ij}\geqslant\beta>0$；当 $(i,j)\notin E_t$ 有 $[\boldsymbol{P}_t]_{ij}=0$；

(iii) $\boldsymbol{P}_t$ 为双随机矩阵。

定义矩阵 $\boldsymbol{Q}_{t\to s}=\boldsymbol{P}_t\boldsymbol{P}_{t-1}\cdots\boldsymbol{P}_{s+1}\boldsymbol{P}_s$，$\forall\, t\geqslant s\geqslant 1$，且 $\Phi_{t\to t}=\boldsymbol{P}_t$，有

$$\left|[\boldsymbol{Q}_{t\to s}]_{ij}-\frac{1}{N}\right|\leqslant\left(1-\frac{\beta}{4N^2}\right)^{\frac{t-s}{B}-2},\quad \forall\, i,j\in V。$$

在线分布式优化可视为多智能体系统和环境/对手的博弈。一般来说，它由以下几步构成

（1）智能体 $i\in V$ 在 $t(t=1,2,\cdots,T)$ 时刻做出决策 $\boldsymbol{x}_i(t)\in X\subseteq R^d$；

（2）环境/对手透露关于损失函数 $l_{i,t}(\boldsymbol{x}):X\to R$ 的信息并遭受损失 $l_{i,t}(\boldsymbol{x}_i(t))$；

（3）智能体 i 与其相邻智能体进行信息交互并基于环境/对手反馈的信息做出下一步决策 $\boldsymbol{x}_i(t+1)$。

一般地，从决策空间角度而言，可分为凸集约束和不等式约束。需要指出的是，本文所讨论的算法都假设决策空间具有有限直径，即 $\max_{\boldsymbol{x},\boldsymbol{y}\in X}\|\boldsymbol{x}-\boldsymbol{y}\|_2\leqslant D$。从信息反馈角度而言，可分为两种经典反馈方式：

（1）完全信息反馈[13,14]：智能体 $i\in V$ 在 t 时刻做完决策后，环境/对手将损失函数 $l_{i,t}(\boldsymbol{x})$ 的完全信息反馈给智能体 i。因此，在算法设计时可以使用损失函数 $l_{i,t}(\boldsymbol{x})$ 的梯度甚至 Hessian 矩阵，即 $\nabla^2 l_{i,t}(\boldsymbol{x})$。

（2）（Bandit 信息反馈[15-17]：智能体 $i\in V$ 在 t 时刻做完决策后，环境/对手只将决策点 $\boldsymbol{x}_i(t)$（或决策点附近 $\tilde{\boldsymbol{x}}_i(t)$）上产生的损失值 $l_{i,t}(\boldsymbol{x}_i(t))$（$l_{i,t}(\tilde{\boldsymbol{x}}_i(t))$）反馈给智能体 i。因此，在算法设计时一般采用以下两种梯度估计器

（1）单点梯度估计器（One-Point Gradient Estimator）[15]

$$g_{i,t}^{OP}=\frac{d}{\varepsilon_t}l_{i,t}(\boldsymbol{x}_i(t)+\varepsilon_t\boldsymbol{u}_i(t))\boldsymbol{u}_i(t)，$$

(2) **两点梯度估计器**(Two-Point Gradient Estimator)[16,17]

$$g_{i,t}^{TP}=\frac{d}{2\varepsilon_t}(l_{i,t}(\boldsymbol{x}_i(t)+\varepsilon_t\boldsymbol{u}_i(t))-l_{i,t}(\boldsymbol{x}_i(t)-\varepsilon_t\boldsymbol{u}_i(t)))\boldsymbol{u}_i(t)\ ,$$

式中 $\varepsilon_t>0$ 为探索参数，$\boldsymbol{u}_i(t)\in R^d$ 为服从某种概率分布的随机探索向量。

2.2 性能指标

2.2.1 **静态遗憾**。一般用静态遗憾来评价在线分布式优化算法性能的优劣。它的定义为：网络中任意智能体 $i\in V$ 的决策序列 $\{\boldsymbol{x}_i(1),\boldsymbol{x}_i(2),\cdots,\boldsymbol{x}_i(T)\}$ 在所有智能体所有时刻的损失函数上 $\{l_{j,1}(\boldsymbol{x}),l_{j,2}(\boldsymbol{x})\cdots,l_{j,T}(\boldsymbol{x})\}_{j=1}^{N}$ 产生的损失与最优决策在所有损失函数上产生的损失之间的差值，其数学形式为

$$Reg_i(T)=\sum_{t=1}^{T}\sum_{j=1}^{N}l_{j,t}(\boldsymbol{x}_i(t))-\sum_{t=1}^{T}\sum_{j=1}^{N}l_{j,t}(\boldsymbol{x}_T^*)\ ,$$

式中 $\boldsymbol{x}_T^*=\underset{\boldsymbol{x}\in X}{\operatorname{argmin}}\sum_{t=1}^{T}\sum_{j=1}^{N}l_{j,t}(\boldsymbol{x})$。一般地，要求所设计的在线分布式优化算法的静态遗憾相较于 T 是次线性增长的，即 $\lim\limits_{T\to\infty}\dfrac{Reg_i(T)}{T}=0,\ \forall i\in V$，记为 $Reg_i(T)=o(T)$。

2.2.2 **动态遗憾**。动态遗憾是一种更严格的性能指标[18-21]，它要求每个时刻智能体的决策向量与当前时刻网络所有损失函数的最优决策相比较，其形式为

$$Reg_i^D(T)=\sum_{t=1}^{T}\sum_{j=1}^{N}l_{j,t}(\boldsymbol{x}_i(t))-\sum_{t=1}^{T}\sum_{j=1}^{N}l_{j,t}(\boldsymbol{x}_t^*)\ ,$$

式中 $\boldsymbol{x}_t^*=\underset{\boldsymbol{x}\in X}{\operatorname{argmin}}\sum_{j=1}^{N}l_{j,t}(\boldsymbol{x})$。不难验证，$Reg_i(T)\leqslant Reg_i^D(T)$。

2.2.3 **复合遗憾**。当智能体 $i\in V$ 在 t 时刻的目标函数由损失函数 $l_{i,t}(\boldsymbol{x})$ 和正则项 $r_i(\boldsymbol{x}):X\to R$ 组成时，需要引入复合遗憾来刻画算法的性能，即

$$Reg_i^C(T)=\sum_{t=1}^{T}\sum_{j=1}^{N}(l_{j,t}(\boldsymbol{x}_i(t))+r_j(\boldsymbol{x}_i(t)))-\sum_{t=1}^{T}\sum_{j=1}^{N}(l_{j,t}(\boldsymbol{x}_T^*)+r_j(\boldsymbol{x}_T^*))\ ,$$

式中 $\boldsymbol{x}_T^*=\underset{\boldsymbol{x}\in X}{\operatorname{argmin}}\sum_{t=1}^{T}\sum_{j=1}^{N}(l_{j,t}(\boldsymbol{x})+r_j(\boldsymbol{x}))$。正则项的引入常常用来对解的结构进行控制，比如，通过设定 $r_i(\boldsymbol{x})=c_i\|\boldsymbol{x}\|_1(c_i>0)$ 以得到稀疏解。

2.2.4 **累加约束违反度**。当决策空间是一组不等式约束时，即 $X=\{h_s(\boldsymbol{x})\leqslant 0,s=1,\cdots,M\}$，其中 $h_s(\boldsymbol{x})$ 是凸函数。智能体无法在每一个时刻计算决策向量到决策空间 X 的精确投影。因此，需要引入累加约束违反度(Cumulative Constraint Violation)来刻画所有智能体的决策向量在所有不等式约束上产生的偏差值

$$CCV(T)=\Big[\sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{s=1}^{M}h_s(\boldsymbol{x}_j(t))\Big]_+\ ,$$

式中 $[z]_+=\max\{0,z\}$。CCV 存在正负相抵消的情况，因此考虑一种更严格的指标，即累加绝对约束违反度(Cumulative Absolute Constraint Violation)

$$CACV(T)=\sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{s=1}^{M}[h_s(\boldsymbol{x}_j(t))]_+\ 。$$

相较于凸集决策空间情形，不仅要求所设计的算法的遗憾是次线性增长的，而且累加(绝对)约束违反度也需要是次线性增长的，即 $CCV(T)=o(T)$ 或 $CACV(T)=o(T)$。

3 算法设计

本节回顾现有经典的在线分布式优化算法，并对算法的设计思路和收敛性结果进行总结。

3.1　在线分布式次梯度算法

文献[22]提出了完全信息反馈下的在线分布式次梯度算法，它可视为分布式次梯度法[4]的在线版本，也可视为在线学习中在线梯度下降法[14]的分布式版本

$$\begin{aligned}&\boldsymbol{y}_i(t+1)=\sum_{j=1}^{N}[\boldsymbol{P}]_{ij}\boldsymbol{x}_j(t)-\eta_t\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t)),\\&x_j(t+1)=\mathrm{Proj}_X[\boldsymbol{y}_i(t+1)]。\end{aligned}\tag{1}$$

在算法中，智能体的初始决策设置为 $\boldsymbol{x}_i(1)\in X,\forall i\in V$；$\boldsymbol{P}$ 为权重矩阵；$\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))\in\partial l_{i,t}(\boldsymbol{x}_i(t))$ 为损失函数 $l_{i,t}(\boldsymbol{x})$ 在 $\boldsymbol{x}_i(t)$ 点任意一个次梯度，当损失函数可微时有 $\partial l_{i,t}(\boldsymbol{x}_i(t))=\{\nabla l_{i,t}(\boldsymbol{x}_i(t))\}$；$\eta_t$ 是 t 时刻的学习率。

算法(1)分为两部分：第一部分是一致性，其作用是使智能体之间的决策趋于一致；第二部分是局部在线优化，每个智能体在 t 时刻使用次梯度法对损失函数 $l_{i,t}(\boldsymbol{x})$ 进行优化，并将向量投影至决策空间 X 得到下一步决策 $\boldsymbol{x}_i(t+1)$。文献[22]证明：当损失函数是凸函数时，通过设置学习率为 $\eta_t=O\left(\frac{1}{\sqrt{t}}\right)$ 可以得到 $Reg_i(T)=O(\sqrt{T})$；当损失函数满足强凸条件时，通过设置学习率为 $\eta_t=O\left(\frac{1}{\sigma t}\right)$ 可以得到 $Reg_i(T)=O(\ln(T))$。需要指出的是，文献[22]同时刻画了静态遗憾与网络拓扑之间的关系。

3.2　在线分布式对偶平均算法

文献[23,24]提出了完全信息反馈下的在线分布式对偶平均算法，它是经典的分布式对偶平均算法[12]的在线推广

$$\begin{aligned}&\boldsymbol{z}_i(t+1)=\sum_{j=1}^{N}[\boldsymbol{P}]_{ij}\boldsymbol{x}_j(t)+\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t)),\\&\boldsymbol{x}_i(t+1)=\prod_X^{\psi}(\boldsymbol{z}_i(t+1),\eta_t),\\&\hat{\boldsymbol{x}}_i(t+1)=\frac{1}{t+1}\sum_{s=1}^{t+1}\boldsymbol{x}_i(s)。\end{aligned}\tag{2}$$

在算法中，智能体的初始决策设置为 $\boldsymbol{x}_i(1)\in X,\forall i\in V$；$\prod_X^{\psi}(\cdot)$ 为到集合 X 的近端投影算子，其定义为

$$\prod_X^{\psi}(\boldsymbol{z},\eta)=\underset{\boldsymbol{x}\in X}{\mathrm{argmin}}\left\{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{x}+\frac{1}{\eta}\psi(\boldsymbol{x})\right\},$$

式中 $\psi(\boldsymbol{x}):X\to R$ 是近端函数，满足强凸性，$\psi(\boldsymbol{x})\geqslant 0$ 和 $\psi(0)=0$。文献[23]证明当 $\psi(\boldsymbol{x})$ 有界且选取 $\eta_t=O\left(\frac{1}{\sqrt{t}}\right)$ 时，算法(2)产生的历史平均决策序列 $\{\hat{\boldsymbol{x}}_i(t)\}_{t=1}^{\mathrm{T}}$ 满足 $Reg_i(T)=O\left(\frac{\sqrt{T}}{1-\sigma_2(\boldsymbol{P})}\right)$。

3.3　在线分布式镜面下降算法

文献[25]提出了时变切换拓扑下的离线分布式镜面下降算法，在此基础上，文献[26]考虑了在线分布式复合优化问题并设计在线分布式复合镜面下降算法。下面首先给出在线分布式镜面下降算法的具体形式以及相关理论结果

$$\begin{aligned}&\nabla\Phi(\boldsymbol{y}_i(t))=\nabla\Phi(\boldsymbol{x}_i(t))-\eta\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t)),\\&\boldsymbol{z}_i(t+1)=\underset{\boldsymbol{x}\in X}{\mathrm{argmin}}\,D_{\Phi}(\boldsymbol{x}\parallel\boldsymbol{y}_i(t)),\\&\boldsymbol{x}_i(t+1)=\sum_{j=1}^{N}[\boldsymbol{P}_t]_{ij}\boldsymbol{z}_j(t+1)。\end{aligned}\tag{3}$$

在算法中，智能体的初始决策设置为 $\boldsymbol{x}_i(1)\in X,\forall i\in V$；$\Phi:R^d\to R$ 为距离产生函数，满足可微性和

强凸性；$D_\Phi(\boldsymbol{x}\parallel\boldsymbol{y})=\Phi(\boldsymbol{x})-\Phi(\boldsymbol{y})-\nabla\Phi(\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})$ 为 Bregman 散度。算法(3)前两行亦可以写成

$$\begin{aligned}\boldsymbol{z}_i(t+1)&=\operatorname*{argmin}_{\boldsymbol{x}\in X}D_\Phi(\boldsymbol{x}\parallel\boldsymbol{y}_i(t))=\operatorname*{argmin}_{\boldsymbol{x}\in X}\{\Phi(\boldsymbol{x})-\nabla\Phi(\boldsymbol{y}_i(t))^{\mathrm{T}}\boldsymbol{x}\}\\&=\operatorname*{argmin}_{\boldsymbol{x}\in X}\{\Phi(\boldsymbol{x})-(\nabla\Phi(\boldsymbol{x}_i(t))-\eta\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t)))^{\mathrm{T}}\boldsymbol{x}\}\\&=\operatorname*{argmin}_{\boldsymbol{x}\in X}\{\eta\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))^{\mathrm{T}}\boldsymbol{x}+D_\Phi(\boldsymbol{x}\parallel\boldsymbol{x}_i(t))\}。\end{aligned}$$

下面给出两种经典的距离产生函数选取方案以及相应的算法。

(i) $\Phi(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{x}\|_2^2$，其对应的 Bregman 散度为 $D_\Phi(\boldsymbol{x}\parallel\boldsymbol{y})=\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$。不难证明，在此情形下算法(3)的形式为

$$\begin{aligned}\boldsymbol{z}_i(t+1)&=\mathrm{Proj}_X[\boldsymbol{x}_i(t)-\eta\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))],\\\boldsymbol{x}_i(t+1)&=\sum_{j=1}^N[\boldsymbol{P}_t]_{ij}\boldsymbol{z}_j(t+1),\end{aligned}$$

其形式和在线分布式次梯度算法(1)极为相似，其区别在于一致性算法和在线局部优化算法的先后顺序不同。

(ii) $\Phi(\boldsymbol{x})=\sum_{i=1}^d[\boldsymbol{x}]_i\ln([\boldsymbol{x}]_i)$，决策空间为 $X=\Delta_d=\{\boldsymbol{x}\in R^d:\sum_{i=1}^d[\boldsymbol{x}]_i=1,[\boldsymbol{x}]_i\geqslant 0,i=1,\cdots,d\}$，其对应的 Bregman 散度为 $D_\Phi(\boldsymbol{x}\parallel\boldsymbol{y})=\sum_{i=1}^d[\boldsymbol{x}]_i\ln\left(\frac{[\boldsymbol{x}]_i}{[\boldsymbol{y}]_i}\right)$，又称为 Kullback-Leibler 散度。在此设定下，算法(3)形式为

$$\begin{aligned}[\boldsymbol{z}_i(t+1)]_s&=\frac{[\boldsymbol{x}_i(t)]_s\exp(-\eta[\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))]_s)}{\sum_{j=1}^d[\boldsymbol{x}_i(t)]_j\exp(-\eta[\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))]_j)},s=1,\cdots,d,\\\boldsymbol{x}_i(t+1)&=\sum_{j=1}^N[\boldsymbol{P}_t]_{ij}\boldsymbol{z}_j(t+1),\end{aligned}$$

式中 $\exp(s)=e^s$。

因此，在线分布式次梯度算法可以视为算法(3)的特例。同时，对于一些具有特殊结构的决策空间 X，计算欧几里得投影是极其困难甚至不可行的[27,28]，而在线分布式镜面下降算法可以有效的利用决策空间的几何特性，得到更简单的算法更新方式，大大降低了计算复杂度。结合文献[25]和文献[26]的理论证明结果，不难证明：当损失函数为凸函数时，通过设置学习率为 $\eta=O\left(\frac{1}{\sqrt{T}}\right)$ 可以得到 $Reg_i(T)=O(\sqrt{T})$。

文献[26]进一步地研究了在线分布式复合优化问题并提出了如下完全信息反馈下的在线分布式复合镜面下降算法

$$\begin{aligned}\boldsymbol{z}_i(t+1)&=\operatorname*{argmin}_{\boldsymbol{x}\in X}\left\{\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))^{\mathrm{T}}\boldsymbol{x}+r(\boldsymbol{x})+\frac{1}{\eta}D_\Phi(\boldsymbol{x}\parallel\boldsymbol{x}_i(t))\right\},\\\boldsymbol{x}_i(t+1)&=\sum_{j=1}^N[\boldsymbol{P}_t]_{ij}\boldsymbol{z}_j(t+1)。\end{aligned}\tag{4}$$

在算法中，考虑了所有正则函数相同的情形，即 $r_i(\boldsymbol{x})=r(\boldsymbol{x}),\forall i\in V$。需要注意的是，算法设计中并没有简单的将 $l_{i,t}(\boldsymbol{x})+r(\boldsymbol{x})$ 视为一个整体并对其使用镜面下降算法，因为对 $r(\boldsymbol{x})$ 进行线性化会破坏期望的解的结构，比如，$r(\boldsymbol{x})=c\|\boldsymbol{x}\|_1(c>0)$。文献[26]证明了当选取学习率为 $\eta=O\left(\frac{1}{\sqrt{T}}\right)$ 时，有 $Reg_i^C(T)=O(\sqrt{T})$。在 Bandit 信息反馈下，文献[26]提出了基于两点梯度估计器的在线分布式复合镜面下降算法

$$\begin{aligned}\boldsymbol{z}_i(t+1)&=\underset{\boldsymbol{x}\in(1-\xi)X}{\operatorname{argmin}}\left\{\boldsymbol{g}_{i,t}^{TP}(\boldsymbol{x}_i(t))^{\mathrm{T}}\boldsymbol{x}+r(\boldsymbol{x})+\frac{1}{\eta}D_{\Phi}(\boldsymbol{x}\parallel\boldsymbol{x}_i(t))\right\},\\ \boldsymbol{x}_i(t+1)&=\sum_{j=1}^{N}[\boldsymbol{P}_t]_{ij}\boldsymbol{z}_j(t+1),\end{aligned}\tag{5}$$

式中 $(1-\xi)X=\{(1-\xi)\boldsymbol{x}:\boldsymbol{x}\in X\}$ 为收缩的决策空间，其作用是保证观测点 $\boldsymbol{x}_i(t)\pm\varepsilon_t\boldsymbol{u}_i(t)$ 在决策空间 X 内。值得注意的是，在 Bandit 信息反馈下一般对决策空间 X 做如下进一步的假设

$$B(\boldsymbol{0},r)\subseteq X\subseteq B(\boldsymbol{0},R),0<r\leqslant R<\infty,$$

基于此假设并选取探索参数 $\varepsilon_t=\varepsilon=\xi r=O\left(\frac{1}{\sqrt{T}}\right)$ 和 $\eta=O\left(\frac{1}{d\sqrt{T}}\right)$，有

$$Reg_i^C(T)=\begin{cases}O(d\sqrt{T}),\|\cdot\|=\|\cdot\|_2,\\ O(d^{3/2}\sqrt{T}),\|\cdot\|=\|\cdot\|_p,\end{cases}$$

可见复合遗憾不仅与决策变量维数相关，还与所选取的范数相关，这是 Bandit 信息反馈下在线分布式优化算法的共性。

文献[29]假设时变最优决策 $\boldsymbol{x}_t^*$ 满足如下动态：$\boldsymbol{x}_{t+1}^*=\boldsymbol{A}\boldsymbol{x}_t^*+\upsilon_t$，其中 $\upsilon_t\in R^d$ 为噪声。提出了如下算法

$$\begin{aligned}\boldsymbol{x}_i(t)&=\boldsymbol{A}\hat{\boldsymbol{x}}_i(t),\\ \boldsymbol{y}_i(t)&=\sum_{j=1}^{N}[\boldsymbol{P}]_{ij}\boldsymbol{x}_j(t),\\ \hat{\boldsymbol{x}}_i(t+1)&=\underset{\boldsymbol{x}\in X}{\operatorname{argmin}}\{\eta\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))^{\mathrm{T}}\boldsymbol{x}+D_{\Phi}(\boldsymbol{x}\parallel\boldsymbol{y}_i(t))\},\end{aligned}\tag{6}$$

通过选取 $\eta=\sqrt{\frac{1-\sigma_2(\boldsymbol{P})}{T}}$ 得到 $Reg_i^D(T)=O\left(\sqrt{\frac{C_TT}{1-\sigma_2(\boldsymbol{P})}}\right)$，其中 $C_T=\sum_{t=1}^{T}\|\boldsymbol{x}_{t+1}^*-\boldsymbol{A}\boldsymbol{x}_t^*\|_2$。

3.4　在线分布式 Frank-Wolfe 算法

基于欧几里得距离产生函数的算法涉及点到凸集的欧几里得投影，当决策空间 X 形式复杂或不具有特殊结构时，求解投影问题本身就是一个凸优化问题[30,31]，这无疑对智能体的计算能力提出了挑战。基于此考虑，文献[32]提出了基于 Frank-Wolfe 法（又称条件梯度法）的在线分布式优化算法，其基本思想是将欧氏投影转化为一个线性优化问题：

$$\begin{aligned}F_{i,t}(\boldsymbol{x})&=\eta\boldsymbol{z}_i(t)^{\mathrm{T}}\boldsymbol{x}+\|\boldsymbol{x}-\boldsymbol{x}_i(1)\|_2^2,\\ \upsilon_i(t)&=\underset{\boldsymbol{x}\in X}{\operatorname{argmin}}\{\nabla F_{i,t}(\boldsymbol{x}_i(t))^{\mathrm{T}}\boldsymbol{x}\},\\ \boldsymbol{x}_i(t+1)&=\rho_t\upsilon_i(t)+(1-\rho_t)\boldsymbol{x}_i(t),\\ \boldsymbol{z}_i(t+1)&=\sum_{j=1}^{N}[\boldsymbol{P}]_{ij}\boldsymbol{z}_j(t)+\boldsymbol{g}_{i,t}(\boldsymbol{x}_i(t))。\end{aligned}\tag{7}$$

在算法中，智能体的初始状态设置为 $\boldsymbol{x}_i(1)\in X,\boldsymbol{z}_i(1)=0,\forall i\in V$。算法(7)中的第二步为线性优化，相比于欧氏投影更易于计算。文献[32]证明：当选取学习率 $\eta=O\left(\frac{1}{T^{3/4}}\right)$ 和 $\rho_t=\frac{1}{\sqrt{t}}$ 时，$Reg_i(T)=O(T^{3/4})$，此遗憾与集中式在线 Frank-Wolfe 算法相同。文献[33]则提出了一种基于代理（Surrogate）损失函数的在线分布式 Frank-Wolfe 算法并证明当损失函数满足强凸性时有 $Reg_i(T)=O(T^{2/3}(\ln(T))^{1/3})$。由理论结果可见，在线分布式 Frank-Wolfe 算法的遗憾劣于以上提到的在线分布式次梯度算法和在线分布式镜面下降算法，但算法(7)避免了求解欧氏投影，计算复杂度更低。

3.5　在线分布式原始-对偶次梯度算法

当决策空间是一组不等式约束时，即 $X=\{h_s(\boldsymbol{x})\leqslant0,s=1,\cdots,M\}$，欧氏投影无法计算。因此，算法

设计一般基于对偶域方法。文献[34]引入了正则拉格朗日(Regularized Lagrangian)函数:

$$L_{i,t}(\boldsymbol{x},\lambda)=l_{i,t}(\boldsymbol{x})+\sum_{s=1}^{M}[\lambda]_s h_s(\boldsymbol{x})-\frac{\theta_t}{2}\|\lambda\|_2^2,$$

式中 $\lambda=([\lambda]_1,\cdots,[\lambda]_M)^{\mathrm{T}}\in R^M$ 为拉格朗日乘子,θ_t 为正则参数。文献[34]在 $M=1$ 和决策空间满足假设 $X\subseteq B(0,R)$ 的前提下,设计了一种在线分布式原始-对偶次梯度算法:

$$\begin{aligned}
\hat{\boldsymbol{x}}_i(t)&=\boldsymbol{x}_i(t)-\alpha_t\nabla_x L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t)),\\
\hat{\lambda}_i(t)&=\lambda_i(t)+\beta_t\nabla_\lambda L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t)),\\
\boldsymbol{x}_i(t+1)&=\mathrm{Proj}_{B(0,R)}\Big[\sum_{j=1}^{N}[\boldsymbol{P}]_{ij}\hat{\boldsymbol{x}}_j(t)\Big],\\
\lambda_i(t+1)&=\mathrm{Proj}_{R+}\Big[\sum_{j=1}^{N}[\boldsymbol{P}]_{ij}\hat{\lambda}_j(t)\Big],
\end{aligned}\tag{8}$$

式中 α_t 和 β_t 分别为原始和对偶变量更新的学习率;$\nabla_x L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t))$ 和 $\nabla_\lambda L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t))$ 分别是拉格朗日函数 $L_{i,t}$ 对于原始和对偶变量在点 $(\boldsymbol{x}_i(t),\lambda_i(t))$ 处的次梯度和超梯度

$$\begin{aligned}
\nabla_x L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t))&=\nabla l_{i,t}(\boldsymbol{x}_i(t))+\lambda_i(t)\nabla h(\boldsymbol{x}_i(t)),\\
\nabla_\lambda L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t))&=h(\boldsymbol{x}_i(t))-\theta_t\lambda_i(t)。
\end{aligned}$$

在 $\max\{\|\nabla_x L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t))\|_2,\|\nabla_\lambda L_{i,t}(\boldsymbol{x}_i(t),\lambda_i(t))\|_2\}\leqslant G$ 假设成立的前提下,文献[34]证明:当损失函数满足凸性并选取 $\theta_t=\frac{1}{t^c},\alpha_t=\frac{1}{t^{c+1/2}},\beta_t=\frac{1}{t^{1/2}}$,有 $Reg_i(T)=O(T^{1/2+c})$ 和 $CCV(T)=O(T^{1-c/2})$,其中 $c\in(0,1/2)$;当损失函数满足强凸性并选取 $\theta_t=O\Big(\frac{1}{\sigma t^c}\Big),\alpha_t=\frac{1}{\sigma t},\beta_t=\frac{1}{\theta_t(t+1)}$,有 $Reg_i(T)=O(T^{\bar{c}})$ 和 $CCV(T)=O(T^{1-\bar{c}/2})$,其中 $\bar{c}\in(0,1)$。

文献[35]对算法(8)进行了改进并得到了更优的上界。具体地讲,提出了如下形式的正则拉格朗日函数

$$L_{i,t}(\boldsymbol{x},\lambda)=l_{i,t}(\boldsymbol{x})+\sum_{s=1}^{M}[\lambda]_s[h_s(\boldsymbol{x})]_+-\frac{\theta_t}{2}\|\lambda\|_2^2,$$

并在此基础上设计了如下算法

$$\begin{aligned}
\boldsymbol{y}_i(t)&=\boldsymbol{x}_i(t)-\alpha_t\Big(\nabla l_{i,t}(\boldsymbol{x}_i(t))+\sum_{s=1}^{M}[\lambda_i(t)]_s\partial[h_s(\boldsymbol{x}_i(t))]_+\Big),\\
\boldsymbol{x}_i(t+1)&=\mathrm{Proj}_{B(0,R)}\Big[\sum_{j=1}^{N}[\boldsymbol{P}_t]_{ij}\boldsymbol{y}_j(t)\Big],\\
[\lambda_i(t+1)]_s&=\frac{[h_s(\boldsymbol{x}_i(t+1))]_+}{\theta_t},s=1,\cdots,M,
\end{aligned}\tag{9}$$

需要指出的是,这里考虑了 M 个不等式约束,并且对偶变量无需进行一致性,降低了通信和计算成本。通过选取合适的参数,证明了:当损失函数满足凸性时,$Reg_i(T)=O(T^{\max\{c,1-c\}})$,$CACV(T)=O(T^{1-c/2})$,其中 $c\in(0,1)$;当损失函数满足强凸性时,$Reg_i(T)=O(\ln(T))$,$CACV(T)=O(\sqrt{T\ln(T)})$。与文献[34]相比,在更严格的性能指标下得到了更优的上界。

文献[36,37]研究了耦合不等式约束下的在线分布式优化问题,即 $\sum_{i=1}^{N}h_i(\boldsymbol{x}_i)\leqslant 0$,其中 $\boldsymbol{x}_i\in X_i$,X_i 为智能体 i 的决策空间。基于推和(Push-Sum)算法,提出了一种在线分布式带耦合不等式约束算法,并证明了算法的遗憾和约束违反度上界。

3.6 其它算法

在线分布式优化算法的研究发展迅速,涌现出了一系列代表性的工作,它们分别从网络结构、通信成本、

计算成本、通信时滞、非凸、隐私保护等角度提出了解决方案[38-51]。这里仅列举几例，不再一一赘述。文献[38]提出了一种基于相对状态符号的在线分布式次梯度算法，其优势在于大大降低了智能体之间的通信成本，同时还能保证当损失函数满足凸性时有 $Reg_i(T)=O(\sqrt{T})$ 。文献[39]研究了压缩通信下的在线分布式优化算法及其收敛性分析，对凸损失函数和强凸损失函数分别得到了 $O(\sqrt{T})$ 和 $O(\ln(T))$ 的遗憾，并进一步地证明了单点和两点 Bandit 信息反馈下的在线分布式优化算法的遗憾。文献[45]研究了通信时滞对在线分布式优化算法的影响。文献[45]则研究了非凸性对在线分布式优化算法的影响，引入一种新的性能指标并刻画了算法的动态遗憾。

4　在线分布式优化算法的应用

本节简要介绍在线分布式优化算法的应用。

4.1　分布式机器学习

考虑 K 个独立同分布的样本构成的训练集 $\{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2),\cdots,(\boldsymbol{x}_K,\boldsymbol{y}_K)\}$ ，其中 $\boldsymbol{x}_k\in R^d$ 是第 k 个样本的特征向量，$\boldsymbol{y}_k\in R$ 是第 k 个样本的标签，$k=1,2,\cdots,K$ 。对于样本，可以通过函数来预测其标签，即 $\hat{\boldsymbol{y}}_k=h(\boldsymbol{x}_k,w),k=1,2,\cdots,K$ ，其中 $w\in R^d$ 为可学习的参数。如果将第 k 个样本的真实标签与预测标签之间的误差关系用损失函数 $f_k(\boldsymbol{w})$ 来表示，那么最小化训练集的平均损失函数可以学习到最优的参数 $\boldsymbol{w}^*$ [52]

$$\min_{\boldsymbol{w}\in R^d}\quad\frac{1}{K}\sum_{k=1}^{K}f_k(\boldsymbol{w})\ ,$$

比如，当选取预测函数 $h(\boldsymbol{x}_k,\boldsymbol{w})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_k$ 并采用平方损失函数时，上述问题变为经典的线性回归问题

$$\min_{\boldsymbol{w}\in R^d}\quad\frac{1}{K}\sum_{k=1}^{K}\frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_k-\boldsymbol{y}_k)^2。$$

随着数据库规模的日益增大，在单个处理器上对所有 K 个样本进行训练将不切实际，因此可行的方法是在多处理器上进行分散训练后再进行综合，此即为分布式优化问题。进一步考虑在一些机器学习问题中样本是在线产生的，比如垃圾邮件过滤，而在线分布式优化可以作为工具解决这类问题[53]，其目标函数为

$$\min_{\boldsymbol{w}\in R^d}\quad\frac{1}{K}\sum_{m=1}^{M}\sum_{i=1}^{N}\frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\mathbf{x}_{i,t}-\boldsymbol{y}_{i,t})^2\ ,$$

式中 $K=MN$ 。我们可以利用第 3 节中的算法有效的解决此类问题。

4.2　多智能体 Multi-Armed Bandit 问题

考虑如下学习问题：N 个决策者重复的在 k 个选项或动作中进行选择。每次做出选择之后，每个决策者会得到一定数值的收益，收益由决策者选择的动作决定的平稳概率分布产生。每个决策者的目标是在总的 T 轮迭代内最大化总收益的期望[54]。具体地讲，智能体 i 希望最小化如下遗憾

$$Reg_i(T)=T\mu_1-E\Big[\sum_{t=1}^{T}\sum_{j=1}^{K}\mu_j 1\{a_i(t)=j\}\Big]\ ,$$

式中 μ_j 为第 j 个选项或动作的期望收益，并假设 $\mu_1\geqslant\mu_2\geqslant\cdots\geqslant\mu_k$ ；$a_i(t)$ 为智能体 i 在 t 时刻选择的选项或动作。智能体之间的交互方式由网络拓扑决定，它可以是固定拓扑或切换拓扑。智能体根据邻居的奖励信息进行信息综合，然后根据经典的 MAB 算法做出自己的决策，再将自己的奖励信息反馈给相邻智能体。多智能体 Multi-Armed Bandit 问题在强化学习、推荐系统、信息检索、动态定价、异常检测等诸多领域有着广泛的应用[55,56]。

5　总结与展望

本文从决策空间、性能指标、信息反馈模型等角度对当前在线分布式优化算法的研究热点进行了梳理和

总结,着重对算法的设计思路和收敛性分析结果进行了剖析。近年来,在线分布式优化的研究得到了国内外学者的广泛关注,但相关研究仍处于起步阶段,尚有许多重要的基本问题值得深入研究。一些未来可能的研究方向包括但不限于:

(1) 非凸优化。在很多实际问题中,目标函数往往是非凸的。比如,稀疏回归问题,虽然目标函数是凸函数,而约束条件是非凸的;比如,推荐系统问题常常伴随着非凸秩约束。非凸优化问题往往存在局部解,同时非凸优化问题的定义形式多样、研究方法各异,这使得非凸优化问题比凸优化问题更具有挑战性。因此,如何针对非凸损失函数,定义相应的性能指标并给出相应的在线分布式优化算法设计方案是极具挑战的。

(2) 遗憾下界。目前,现有工作大多致力于推导算法的遗憾上界,而对于此上界是否为最优的讨论极少。而我们知道,算法的遗憾下界决定了算法的收敛性能极限,这对深入研究在线分布式优化算法的收敛性质是极为关键的。因此,借助凸优化理论刻画算法的遗憾下界甚至证明遗憾上界和下界是阶次一致的,是一个值得深入研究的课题。

(3) 自适应学习率。现有部分工作在设计学习率时要么用到总的迭代轮数的信息,要么用到所有损失函数的次梯度的共同范数上界,而算法在更新当前时刻的决策时是无法知道未来损失函数的信息。在分布式环境中,自适应学习率的设计尤为关键,它可以减少智能体之间协调学习率的次数,从而降低通信成本。因此,这些设计都存在局限性,如何设计一类只基于历史信息的自适应学习率是一个值得继续探索的方向。

参 考 文 献

[1] OLFATI-SABER R, MURRAY R. Consensus problems in the networks of agents with switching topology and time delays[J]. IEEE Transactions on Automatic Control, 2004, 49(9): 1520-1533.

[2] REN W, BEARD R. Consensus seeking in multi-agent systems under dynamically changing interaction topologies[J]. IEEE Transactions on Automatic Control, 2005, 50(3): 655-661.

[3] TSITSIKLIS J N, BERTSEKAS D P, ATHANS M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms[J]. IEEE Transactions on Automatic Control, 1986, 31(9): 803-812.

[4] NEDIC A, OZDAGLAR A. Distributed subgradient methods for multi-agent optimization[J]. IEEE Transactions on Automatic Control, 2009, 54(1): 48-61.

[5] 洪奕光, 张艳琼. 分布式优化: 算法设计和收敛性分析[J]. 控制理论与应用, 2014, 31(7): 850-857.

[6] 王龙, 卢开红, 关永强. 分布式优化的多智能体方法[J]. 控制理论与应用, 2019, 36(11): 1820-1833.

[7] 衣鹏, 洪奕光. 分布式合作优化及其应用[J]. 中国科学:数学, 2016, 46(10): 1547-1564.

[8] 谢佩, 游科友, 洪奕光, 等. 网络化分布式凸优化算法研究进展[J]. 控制理论与应用, 2018, 35(7): 918-927.

[9] NEDIC A, OZDAGLAR A, PARRILO P A. Constrained consensus and optimization in multi-agent networks[J]. IEEE Transactions on Automatic Control, 2010, 55(4): 922-938.

[10] NEDIC A, OLSHEVSKY A. Distributed optimization over time-varying directed graphs[J]. IEEE Transactions on Automatic Control, 2015, 60(3): 601-615.

[11] NEDIC A, OLSHEVSKY A. Stochastic gradient-push for strongly convex functions on time-varying directed graphs[J]. IEEE Transactions on Automatic Control, 2016, 61(12): 3936-3947.

[12] DUCHI J C, AGARWAL A, WAINWRIGHT M J. Dual averaging for distributed optimization: convergence analysis and network scaling [J]. IEEE Transactions on Automatic Control, 2012, 57(3):592-606.

[13] ZINKEVICH M. Online convex programming and generalized infinitesimal gradient ascent[C]// Proceedings of the Twentieth International Conference on Machine Learning, August 21-24, Washington DC, USA, 2003: 421-422.

[14] HAZAN E. Introduction to online convex optimization[M]. Boston: Foundations and Trends in Machine Learning, 2016, 2(3-4): 157-325.

[15] FLAXMAN A D, KALAI A T, MCMAHAN B H. Online convex optimization in the bandit setting: gradient descent without a gradient

[C]// Proceedings of the 16h Annual ACM-SIAM Symposium on Discrete Algorithms, Vancouver, British Columbia, 2005: 385-394.

[16] AGARWAL A, DEKEL O, XIAO L. Optimal algorithms for online convex optimization with multi-point bandit feedback[C]// The 23rd Annual Conference on Learning Theory, June 27-29, Haifa, Israel, 2010: 28-40.

[17] SHAMIR O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback[J]. Journal of Machine Learning Research, 2017, 18(52): 1-11.

[18] HALL E, WILLETT R. Online convex optimization in dynamic environments[J]. IEEE Journal of Selected Topics in Signal Processing, 2015, 9(4): 647-662.

[19] BESBES O, GUR Y, ZEEVI A. Non-stationary stochastic optimization[J]. Operations Research, 2015, 63(5): 1227-1244.

[20] JADBABAIE A, RAKHLIN A, SHAHRAMPOUR S, et al. Online optimization: competing with dynamic comparators[C]// Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, May 9-12, San Diego, CA, 2015: 398-406.

[21] SHAHARAMPOUR S, JADBABAIE A. An online optimization approach for multi-agent tracking of dynamic parameters in the presence of adversarial noise[C]// American Control Conference (ACC). May 24-26, Seattle, Washington, USA, 2017: 3306-3311.

[22] YAN F, SUNDARAM S, VISHWANATHAN S V N, et al. Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(11): 2483-2493.

[23] HOSSEINI S, CHAPMAN A, MESBAHI M. Online distributed optimization via dual averaging[C]// Proceedings of the 52nd IEEE Conference on Decision and Control, December 10-13, Florence, Italy, 2013: 1484-1489

[24] HOSSEINI S, CHAPMAN A, MESBAHI M. Online distributed convex optimization on dynamic networks[J]. IEEE Transactions on Automatic Control, 2016, 61(11): 3545-3550.

[25] YUAN D, HONG Y, HO D W C, et al. Optimal distributed stochastic mirror descent for strongly convex optimization[J]. Automatica, 2018, 90:196-203.

[26] YUAN D, HONG Y, HO D W C, et al. Distributed mirror descent for online composite optimization[J]. IEEE Transactions on Automatic Control, 2020, 66(2): 714-729.

[27] BOYD S P, VANDENBERGHE L. Convex optimization[M].Cambridge: Cambridge university press, 2004.

[28] BUBECK S. Convex optimization: Algorithms and complexity[M].Boston: Foundations and Trends in Machine Learning, 2015.

[29] SHAHRAMPOUR S, JADBABAIE A. Distributed online optimization in dynamic environments using mirror descent[J]. IEEE Transactions on Automatic Control, 2018, 63(3): 714-725.

[30] HAZAN E, KALE S. Projection-free online learning[C]// Proceedings of the 29th International Conference on Machine Learning, June 26-July 1, Edinburgh, Scotland, 2012: 521-528.

[31] JAGGI M. Revisiting frank-wolfe: projection-free sparse convex optimization[C]// Proceedings of the 30th International Conference on Machine Learning, June 17-19, Atlanta, Georgia, USA, 2013: 427-435.

[32] ZHANG W, ZHAO P, ZHU W, et al. Projection-free distributed online learning in networks[C]// Proceedings of the 34th International Conference on Machine Learning, August 6-11, Sydney, Australia, 2017: 4054-4062.

[33] WAN Y, WANG G, TU W W, et al. Projection-free distributed online learning with sublinear communication complexity[J]. Journal of Machine Learning Research, 2022, 23(172): 1-53.

[34] YUAN D, HO D W C, JIANG G P. An adaptive primal-dual subgradient algorithm for online distributed constrained optimization[J]. IEEE Transactions on Cybernetics, 2018, 48(11): 3045-3055.

[35] YUAN D, PROUTIERE A, SHI G. Distributed online optimization with long-term constraints[J]. IEEE Transactions on Automatic Control, 2022, 67(3): 1089-1104.

[36] LI X, YI X, XIE L. Distributed online optimization for multi-agent networks with coupled inequality constraints[J]. IEEE Transactions on Automatic Control, 2021, 66(8): 3575-3591.

[37] YI X, LI X, YANG T, et al. Distributed bandit online convex optimization with time-varying coupled inequality constraints[J]. IEEE Transactions on Automatic Control, 2021, 66(10): 4620-4635.

[38] CAO X, BAŞAR T. Decentralized online convex optimization based on signs of relative states[J]. Automatica, 2021, 129: 109676.

[39] TU Z, WANG X, HONG Y, et al. Distributed online convex optimization with compressed communication[C]// Advances in Neural Information Processing Systems 35 (NeurIPS 2022), 2022: 34492-34504.

[40] SOOMIN L, NEDIC, RAGINSKY M. Stochastic dual averaging for decentralized online optimization on time-varying communication

graphs[J]. IEEE Transactions on Automatic Control, 2017, 62(12): 6407-6414.

[41] LEE S, NEDIC A, RAGINSKY M. Stochastic dual averaging for decentralized online optimization on time-varying communication graphs [J]. IEEE Transactions on Automatic Control 2017, 62(12): 6407-6414.

[42] XIONG Y, LI X, YOU K, et al. Distributed online optimization in time-varying unbalanced networks without explicit subgradients[J]. IEEE Transactions on Signal Processing, 2022, DOI: 10.1109/TSP.2022.3194369.

[43] PANG Y, HU G. N. Hayashi. Randomized gradient-free distributed online optimization via a dynamic regret analysis[J]. IEEE Transactions on Automatic Control, 2023, DOI: 10.1109/TAC.2023.3237975.

[44] CAO X, BASAR T. Decentralized online convex optimization with feedback delays[J]. IEEE Transactions on Automatic Control, 2022, 67(6): 2889-2904.

[45] LU K, WANG L. Online distributed optimization with nonconvex objective functions via dynamic regrets[J]. IEEE Transactions on Automatic Control[J], 2023, DOI: 10.1109/TAC.2023.3239432.

[46] LEI J, YI P, HONG Y, et al. Online convex optimization over Erdos-Rényi random networks[C]// Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020: 15591-15601.

[47] MATEOS-NÚNEZ D, CORTÉS J. Distributed online convex optimization over jointly connected digraphs[J]. IEEE Transactions on Network Science and Engineering, 2014, 1(1): 23-37.

[48] LU K, JING G, WANG L. Online distributed optimization with strongly pseudoconvex-sum cost functions[J]. IEEE Transactions on Automatic Control, 2019, 65(1): 426-433.

[49] LU K, WANG L. Online distributed optimization with nonconvex objective functions: sublinearity of first-order optimality condition-based regret[J]. IEEE Transactions on Automatic Control, 2022, 67(6): 3029-3035.

[50] AKBARI M, GHARESIFARD B, LINDER T. Distributed online convex optimization on time-varying directed graphs[J]. IEEE Transactions on Control of Network Systems, 2017, 4(3): 417-428.

[51] XIONG Y, XU J, YOU K, et al. Privacy-preserving distributed online optimization over unbalanced digraphs via subgradient rescaling [J]. IEEE Transactions on Control of Network Systems, 2020, 7(3): 1366-1378.

[52] **史加荣, 王丹, 尚凡华, 等. 随机梯度下降算法研究进展**[J]. **自动化学报**, 2021, 47(9): 2103-2119.

[53] YUAN D, PROUTIERE A, SHI G. Distributed online linear regressions[J]. IEEE Transactions on Information Theory, 2021, 67(1): 616-639.

[54] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 2018.

[55] SHAHRAMPOUR S, RAKHLIN A, JADBABAIE A. Multi-armed bandits in multi-agent networks[C]// 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 2786-2790.

[56] VIAL D, SHAKKOTTAI S, SRIKANT R. Robust multi-agent multi-armed bandits[C]// Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing. 2021: 161-170.

(责任编辑:刘庆松)