DOI:10.13878/j.cnki.jnuist.2018.06.001

张家绮[1,2]　游科友[1,2]

基于相对状态符号信息的分布式优化算法

摘要

针对网络化多智能体的分布式优化问题,本文讨论一种只利用邻居相对状态的符号信息的分布式算法.该算法不要求与图相关的权重矩阵是双随机矩阵.首先利用优化理论中的惩罚函数法解释该算法,然后分析算法在静态图上的收敛性以及收敛速度.与现有使用邻居相对状态的完整信息的分布式梯度下降算法相比,所提算法的收敛速度并没有本质上降低.另一方面,将所提算法扩展到确定性和随机性的时变图上,并给出相应的收敛性结论.最后,通过数值仿真实验验证算法的有效性.

关键词

分布式优化;多智能体网络;相对状态符号;惩罚函数法;次梯度方法

中图分类号 TP13

文献标志码 A

收稿日期 2018-08-29

资助项目 国家自然科学基金(61722308);国家重点研发计划(2017YFC0805310)

作者简介

张家绮,男,博士生,研究方向为分布式优化、学习及应用.zjq16@mails.tsinghua.edu.cn

游科友(通信作者),男,博士,副教授,博士生导师,研究方向为网络化系统的学习、优化与控制及其应用.youky@tsinghua.edu.cn

1 清华大学 自动化系,北京,100084

2 清华大学 北京信息科学与技术国家研究中心,北京,100084

0 引言

近些年来,多智能体系统中的分布式优化问题得到了越来越多的关注.分布式优化是指多智能体系统中的所有智能体协作寻找一个全局目标函数的最小值点,这个全局目标函数是每个智能体的局部目标函数之和.分布式优化问题的一个显著特点是每个智能体只知道其对应的局部目标函数,因此智能体之间必须合作.通常用图来表示智能体之间的通信拓扑结构.分布式优化问题在实际中有很多应用,比如 AUV(自主式水下机器人)的队形控制[1]、大规模机器学习[2-4]以及传感器网络中的分位点回归问题[5]等.

求解这类问题的主要算法(参考文献[6-8]以及文中所引用的参考文献)通常由两部分组成.第一部分是为了使所有智能体的状态趋于一致,我们称之为一致项,它通常基于现有的一致性算法[9].第二部分是使达到一致后智能体的状态收敛到全局目标函数的最小值点,这一部分可利用的信息通常是每个智能体局部目标函数的值及其次梯度,因此次梯度算法被广泛采用.为了实现所有智能体状态一致的目标,现有主流算法要求每个智能体能够获得它的邻居智能体的状态的精确值[8-9]或者量化值[10-11].但是,有些情况下智能体只能获得它和它邻居之间粗略的相对状态信息.比如,考虑这样的场景,多个智能体(机器人)在二维平面上运动,每个智能体只知道邻近智能体是在它的左边或者右边、上边或者下边,而不是准确的相对位置信息.因此,这种情形下每个智能体在每个坐标轴下只能从邻居那里获得相对位置的符号信息.这与智能体的量化状态工作信息[11]显著不同,这些工作利用智能体的状态量化值而不是精确值.此外,也有别于利用量化梯度信息的研究[12].总之,大部分已有的工作,都不能处理只使用相对状态符号信息的情况.除此之外,本文所提算法的另一个优点是不要求智能体之间的交互矩阵满足双随机性,而只要求是对称的.注意到双随机矩阵的假设在已有算法中很常见[6,13-14],但是在实际的分布式优化问题中可能难以满足.比如,常用的构造双随机矩阵的 Metropolis 方法需要每个智能体都知道邻居的出度,这在实际应用中可能无法满足,特别是当智能体之间的通信拓扑是时变的情形.

设计基于相对状态符号信息的算法往往涉及到非线性分析,因此和大多数利用代数图论作为分析工具的算法有本质不同.在文献[15]中,作者提出了一种只利用 1 比特相对状态信息的一致性算法,

但并没有考虑优化问题.类似的算法可以用来计算数据的中位点[16].文献[17]中的算法和本文最为接近,但它是一个连续时间下的算法,并且采用了和本文完全不同的分析方法,我们将在后文进行更深入的比较.

实际上,现有使用相对状态符号信息的工作都是考虑连续时间下的情况.但是,离散时间下的算法是有必要研究的,因为很多算法都要求不同智能体之间能够进行通信并需要对智能体进行控制,而目前实际应用中的通信和控制信号往往是离散的.除此之外,离散时间下的算法实施起来也更简单.另一方面,连续时间下的算法往往基于李雅普诺夫理论,通过构造李雅普诺夫函数进行分析,而离散时间算法中引入的步长通常不能保证李雅普诺夫函数单调不增.因此,李雅普诺夫方法很难应用在离散时间算法的分析中,需要新的分析工具来研究这种非线性离散时间下的算法,这也是本文关心的问题.

具体来说,本文讨论离散时间下只利用相对状态符号信息的分布式优化算法.和大多数已有方法不同,本文的分析基于优化理论而不是代数图论或李雅普诺夫理论.采用优化理论进行分析有两点潜在的好处.首先,目前主要研究思路是先提出一个算法,然后找到一个李雅普诺夫函数来证明算法的稳定性和收敛性.与之相比,基于优化理论得到的算法在直观上更容易被理解和接受,因为我们的算法实质上是在最小化一个经过精心设计的目标函数.其次,丰富的优化理论使得所提算法更容易扩展和改进.比如,我们在时变图上得到的算法就是在静态图上算法的一个直接扩展.具体来说,我们将静态图上的算法扩展到了确定性时变图和随机时变图上,其中确定性时变图能够用来表示实际应用中智能体间时变的通信拓扑,而随机时变图则可以描述 gossip 网络和通信网络中的随机丢包等现象.基于优化理论,本文分别采用了增量式梯度方法和随机梯度法来设计和分析确定性时变图和随机时变图上的算法.

本文的主要结论如下:对于一个静态图,证明了所有智能体的状态在所提算法下都会渐近收敛到全局目标函数的同一个最小值点,并且收敛速度相对于已有算法没有本质上的降低;对于确定性时变图,引入了一致连通的概念,并且证明了算法在一致连通图下的收敛性;对于随机时变图,考虑了一种特殊的情况——随机激活图,并且在概率 1 的意义下证明了算法在随机激活图下的收敛性.

本文其余部分的结构如下:第 1 节介绍一些预备知识并且给出分布式优化问题的定义;第 2 节介绍所提出的离散时间下基于相对状态符号信息的分布式优化算法;第 3 节包含算法在静态图上的收敛性和收敛速度分析;第 4 节研究算法在一致连通图和随机激活图上的收敛性;在第 5 节中利用所提算法进行数值仿真并将仿真结果和理论结果进行了比较;第 6 节对全文进行总结.

本文用到的一些符号定义如下:a,$\boldsymbol{a}$,$\boldsymbol{A}$ 以及 $\mathcal{A}$ 分别表示标量、向量、矩阵以及集合.$\boldsymbol{a}^{\mathrm{T}}$ 和 $\boldsymbol{A}^{\mathrm{T}}$ 分别表示 $\boldsymbol{a}$ 和 $\boldsymbol{A}$ 的转置.$[\boldsymbol{A}]_{ij}$ 表示 $\boldsymbol{A}$ 的第 i 行和第 j 列的元素.$\mathbf{R}$ 表示实数集,$\mathbf{R}^n$ 表示 n 维实数向量集.$\mathbf{1}$ 表示全为 1 的向量,其维数由上下文决定.$\|\cdot\|_1$,$\|\cdot\|_2$ 以及 $\|\cdot\|_\infty$ 分别表示向量或矩阵的 l_1 范数、l_2 范数和 l_∞ 范数.定义符号函数如下:

$$\operatorname{sgn}(x)=\begin{cases}1, & x\geqslant 0,\\ -1, & \text{其他}.\end{cases}$$

$\nabla f(x)$ 表示 $f(x)$ 在 x 处的任意一个次梯度.也就是说,$\nabla f(x)$ 可以是任意满足下列不等式的向量:

$$f(y)\geqslant f(x)+(y-x)^{\mathrm{T}}\nabla f(x),\forall y\in\mathbf{R},$$

$f(x)$ 在 x 处的次微分是 $f(x)$ 在 x 处所有次梯度的集合.如果 $f(x)$ 在 x 处可微,那么次微分是一个单点集,只包含 $f(x)$ 在 x 处的梯度.

我们将 $\inf_{x\in\mathbb{R}^n}f(\boldsymbol{x})$ 叫做 $f(\boldsymbol{x})$ 的最优(小)值,将集合 $\arg\inf_{x\in\mathbb{R}^n}f(\boldsymbol{x})$ 中的任意元素叫做 $f(\boldsymbol{x})$ 的一个最优(小)值点.

本文中上标通常用来表示数列的索引,例如,x^k 表示数列 $\{x^k\}$ 中的第 k 个数.

1 问题描述

这一节介绍图论中的一些基础知识,并且引出多智能体系统中的分布式优化问题.

1.1 代数图的基本概念

一个图或者网络可以用 $\mathcal{G}=(\mathcal{V},\mathcal{E})$ 来表示,其中 $\mathcal{V}=\{1,\cdots,n\}$ 是节点的集合,$\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ 是边的集合.$\mathcal{N}_i=\{j\in\mathcal{V}\mid(i,j)\in\mathcal{E}\}$ 表示节点 i 的邻居(节点)的集合.$\boldsymbol{A}=[a_{ij}]$ 表示 $\mathcal{G}$ 的邻接矩阵,其中 $a_{ij}>0$ 当且仅当节点 i 和 j 之间存在一条边相连;否则 $a_{ij}=0$.如果 $\boldsymbol{A}$ 是对称矩阵,那么称图 $\mathcal{G}$ 为无向图.本文主要讨论无向图.

如果一个图是时变的,那么用 $\mathcal{G}^k=(\mathcal{V},\mathcal{E}^k,\boldsymbol{A}^k)$ 来表示其在第 k 时刻的图.$\mathcal{G}^{k1}\cup\mathcal{G}^{k2}$ 表示图 $(\mathcal{V},\mathcal{E}^{k1}\cup\mathcal{E}^{k2},\boldsymbol{A}^{k1}+\boldsymbol{A}^{k2})$,$\mathcal{N}_i^k=\{j\in\mathcal{V}\mid(i,j)\in\mathcal{E}^k\}$

表示节点 i 在第 k 时刻的邻居的集合.

图的关联矩阵 $\boldsymbol{B} \in \mathbf{R}^{n\times m}$ 定义如下:

$$\boldsymbol{B}_{ie}=\begin{cases}1, & \text{如果节点 } i \text{ 是边 } e \text{ 的起始节点},\\ -1, & \text{如果节点 } i \text{ 是边 } e \text{ 的终止节点},\\ 0, & \text{其他}.\end{cases}$$

对于任意向量 $\boldsymbol{x}=[x_1,\cdots,x_n]^{\mathrm{T}}$,有

$$\boldsymbol{b}_e^{\mathrm{T}}\boldsymbol{x}=x_i-x_j,$$

其中 $\boldsymbol{b}_e, e\in\mathcal{E}$ 表示 $\boldsymbol{B}$ 的第 e 列,i 和 j 分别是边 e 的起始节点和终止节点.

图的一条路径是指从一个节点到另一个节点所经过的边的集合. 如果图中任意两个节点之间都至少存在一条路径,那么这个图是连通的. 为了表示图的连通强度,我们引入 l- 连通图的概念.

定义 1(l- 连通图)　如果一个连通图在去掉任意 l 条边后仍然是连通的,那么这个连通图叫做 l- 连通图.

显然,一个连通图至少是 1- 连通图,并且 l- 连通图的每个节点至少和 l 条边相连.

1.2　分布式优化问题

本文目标是在多智能体系统中分布式地求解以下优化问题,并且要求每个智能体只用到其与邻居相对状态的符号信息:

$$\underset{x\in\mathbf{R}}{\text{minimize}}\, f(x):=\sum_{i=1}^{n} f_i(x), \tag{1}$$

其中,每个子目标函数 $f_i(x), i\in\mathcal{V}$ 是连续的凸函数,但是可能不可微. 此外,只有节点 i 能够获得 $f_i(x)$ 的具体形式. n 为智能体的个数,这里只考虑 $n>1$ 的一般情况. 我们首先做出一个假设:

假设 1(最小值点的存在性以及有界次梯度)

(a) 最小值点的集合是非空的,即对于任意 $x^*\in\mathcal{X}^*$,有 $f^*:=f(x^*)=\inf_{x\in\mathbf{R}} f(x)$.

(b) 存在某个正数 c 使得

$$|\nabla f_i(x)|\leqslant c, \forall_i\in\mathcal{V}, x\in\mathbf{R}. \tag{2}$$

假设 1 在相关文献中很常见,请参考文献[2,6]. 特别地,假设 1 的第二部分经常被用来保证次梯度算法的收敛性[6],并且在优化变量 x 被限制在某个紧集中时成立.

2　静态图上的分布式优化算法

这一节给出只利用相对状态符号信息的离散时间分布式优化算法,并且用优化理论中的惩罚函数法来直观地解释这个算法.

这一节主要考虑静态图的情况,但是所得结论在之后几节的时变图分析中也很重要.

2.1　分布式优化算法

在静态图上求解分布式优化问题的离散时间算法流程如算法 1 所示.

文献[17]给出了算法 1 的一种连续时间形式,并且通过非光滑分析方法[18]证明了其收敛性. 为了保证算法的有效性和收敛性,λ 和 ρ^k 的选择至关重要. 而对于离散时间而言,这两个参数的选择方法和连续时间下的算法[17]完全不同,将在第 2.2 节中详细说明.

与广泛使用的分布式梯度下降算法(DGD,文献参考[6])

$$x_i^{k+1}=x_i^k+\sum_{j\in\mathcal{N}_i}\bar{a}_{ij}(x_j^k-x_i^k)-\rho^k\nabla f_i(x_i^k) \tag{3}$$

相比,算法 1 至少有两个优点. 首先,在算法 1 中每个节点 i 只需要利用其与邻居相对状态的符号,而不是准确的相对状态. 在某些只有相对状态符号信息可以利用的特殊情况中,这是一个很大的优势. 其次,算法 1 不要求邻接矩阵 $\boldsymbol{A}^k$ 是双随机矩阵,而 DGD 中的邻接矩阵 $[\widetilde{\boldsymbol{A}}^k]_{ij}:=\widetilde{A}_{ij}^k$ 则必须是双随机的,这使得其在实际中的应用范围受到限制.

注 1　当 x 是向量时,通过将符号函数 sgn(·) 应用到相对状态向量的每个元素上,算法 1 依然有效. 在这种情况下,本文中的所有结论略微修正后依然成立.

算法 1:基于相对状态符号信息的分布式优化算法(静态图)

1) 初始化. 对于每个节点 $i\in\mathcal{V}$,令 $x_i^0=0$.

2) 开始循环.

3) 信息交换. 每个节点 $i\in\mathcal{V}$ 通过某种方式获得其与邻居节点 $j\in\mathcal{N}_i$ 的相对状态信息的符号,并按照如下公式计算 r_i^k:

$$r_i^k=\sum_{j\in\mathcal{N}_i} a_{ij}\mathrm{sgn}(x_j^k-x_i^k).$$

4) 状态更新. 按照如下方式更新每个节点 $i\in\mathcal{V}$ 的自身状态:

$$x_i^{k+1}=x_i^k+\rho^k(\lambda\cdot r_i^k-\nabla f_i(x_i^k)),$$

其中 λ 和 ρ^k 的选择由定理 2 给出,$\nabla f_i(x_i^k)$ 为 $f_i(x)$ 在 x_i^k 处的任意次梯度.

5) 令 $k=k+1$.

6) 终止条件. 如果一个预先设定的终止条件(如最大迭代次数)得到满足,则算法终止.

2.2 算法1的惩罚函数解释

这一小节将利用惩罚函数法来解释算法1,并且说明算法1实质上是原问题对应的惩罚问题的次梯度方法.

首先,原问题(1)可以写成如下形式:

$$\begin{aligned}&\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad g(\boldsymbol{x}):=\sum_{i=1}^{n}f_i(x_i)\ ,\\&\text{subject to} \quad x_i=x_j,\forall i,j\in\{1,\cdots,n\},\end{aligned}\tag{4}$$

其中$\boldsymbol{x}=\{x_1,\cdots,x_n\}^{\mathrm{T}}$.显然,上述问题(4)的最优值仍然是$f^*$,并且其最优值点集合是$\{x^*\mathbf{1}\mid x^*\in\mathcal{X}^*\}$.

定义一个惩罚函数如下所示:

$$h(\boldsymbol{x})=\frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}a_{ij}\mid x_i-x_j\mid\ ,\tag{5}$$

如果图$\mathcal{G}$是连通的,那么$h(\boldsymbol{x})=0$等价于$x_i=x_j,\forall i,j\in\{1,\cdots,n\}$.因此,可以得到问题(4)对应的一个惩罚问题,如下所示:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \tilde{f}_\lambda(\boldsymbol{x}):=g(\boldsymbol{x})+\lambda h(\boldsymbol{x})\ ,\tag{6}$$

其中$\lambda>0$是惩罚系数.

接下来说明算法1恰好是上述惩罚问题(6)在步长ρ^k下的次梯度方法.首先注意到对于任意$x\in\mathbf{R}$,$\mathrm{sgn}(x)$是$\mid x\mid$的一个次梯度.因此,由(5)可知,$h(\boldsymbol{x})$的一个次梯度$\nabla h(\boldsymbol{x})=[\nabla h(\boldsymbol{x})_1,\cdots,\nabla h(\boldsymbol{x})_n]^{\mathrm{T}}$可以通过如下公式给出:

$$\nabla h(\boldsymbol{x})_i=\sum_{j\in\mathcal{N}_i}a_{ij}\mathrm{sgn}(x_i-x_j),i\in\mathcal{V}.$$

同样地,$g(\boldsymbol{x})$的一个次梯度$\nabla g(\boldsymbol{x})=[\nabla g(\boldsymbol{x})_1,\cdots,\nabla g(\boldsymbol{x})_n]^{\mathrm{T}}$也可以通过以下方式给出:

$$\nabla g(\boldsymbol{x})_i=\nabla f_i(x_i).$$

于是,$g(\boldsymbol{x})$的次梯度的第i个元素可以写成以下形式:

$$\nabla\tilde{f}_\lambda(\boldsymbol{x})_i=\lambda\sum_{j\in\mathcal{N}_i}a_{ij}\mathrm{sgn}(x_i-x_j)+\nabla f_i(x_i),i\in\mathcal{V}.$$

最后,惩罚问题(6)的次梯度方法如下所示:

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\rho^k\nabla\tilde{f}(\boldsymbol{x}^k),$$

这恰好就是将算法1写成向量更新后的形式.由文献[19]可知,如果步长ρ^k选取得合适,那么上述次梯度方法生成的$\{\boldsymbol{x}^k\}$一定会收敛到惩罚问题(6)的一个最小值点.

如果任意选择惩罚系数λ为某个正数(不包括正无穷),问题(4)和惩罚问题(6)一般而言不等价.尽管如此,我们发现在一定条件下,λ如果严格大于某个下界,那么这两个问题将变得等价.为此,定义

$$\bar{x}=\frac{1}{n}\mathbf{1}^{\mathrm{T}}\boldsymbol{x},$$

$$v(\boldsymbol{x})=\max_i(x_i)-\min_i(x_i),\tag{7}$$

并且令$a_{\min}^{(l)}$表示图中权重最小的l条边的权重和,即,

$$a_{\min}^{(l)}=\sum_{e=1}^{l}a_{(e)}\ ,\tag{8}$$

其中$a_{(1)},a_{(2)},\cdots$是所有边的权重$a_{ij},(i,j)\in\varepsilon$按照升序排序后的序列.

定理1(惩罚系数的下界,文献[20]) 假设l成立,多智能体系统对应的图是l-连通图.如果惩罚系数λ满足

$$\lambda>\underline{\lambda}:=\frac{nc}{2a_{\min}^{(l)}},\tag{9}$$

其中c和$a_{\min}^{(l)}$分别由式(2)和式(8)给出,那么:

(a) 原优化问题(1)和惩罚问题(6)等价,即惩罚问题(6)的最优值和最优值点的集合分别为f^*和$\tilde{\mathcal{X}}^*=\{x^*\mathbf{1}\mid x^*\in\mathcal{X}^*\}$.

(b) 对于任意向量$\boldsymbol{x}\notin\{\alpha\mathbf{1}\mid\alpha\in\mathbf{R}\}$,有

$$\|\nabla\bar{f}_\lambda(\boldsymbol{x})\|_\infty\geqslant\frac{2\lambda a_{\min}^{(l)}}{n}=c.$$

假设1(b)可以进一步放松以处理目标函数具有无界次梯度的情况,比如二次函数等,具体参考文献[20].定理1给出了使问题(4)和惩罚问题(6)等价的充分条件,从而我们可以只关注惩罚问题(6).注意,尽管惩罚函数法已经在优化理论中被广泛研究[19],定理1的结论仍然具有显著的意义.众所周知,惩罚问题(6)的最优值和问题(4)的最优值之间的差距随着λ的增大而减小,但是这个结论并不能保证存在一个有限的λ使得两者差距为0.而太大的λ可能会使算法1的优化过程波动相对较大.

注2 这里有必要指出,当x为多维向量时,定理1中的式(9)仍然成立.不过这种情况下需要将假设1(b)修改为对于所有i和向量$\boldsymbol{x}$,$\|\nabla f_i(\boldsymbol{x})\|\leqslant c$都成立.

根据对偶理论[19],惩罚系数λ的下界可以通过问题(4)的拉格朗日乘子得到.然而,获得优化问题的拉格朗日乘子往往需要求解对偶问题,而定理1给出了惩罚系数λ的一个显式下界,并通过网络的大小和连通度显式表出.此外,定理1中的下界比文献[16-17]中的类似下界更紧.

实际上,定理1给出的下界在某些情况下是一个紧的下界.下面例子就验证了这种情况.

例1[20] 考虑如图1b所示的一个无向图,其中所有边的权重都为1,即对于所有$(i,j)\in\mathcal{V},a_{ij}=1$.

令 $f_1(x)=|x|$,$f_2(x)=|x-2|$,$f_3(x)=|x-4|$,$f_4(x)=|x-6|$ 以及 $f(x)=\sum_{i=1}^{4}f_i(x)$.易知 $f(x)$ 的最优值是 8 并且最优值点集合是区间[2,4].由(6)可知,原问题对应的惩罚问题是

$$\tilde{f}_\lambda(\boldsymbol{x})=|x_1|+|x_2-2|+|x_3-4|+|x_4-6|+\lambda(|x_1-x_2|+|x_2-x_3|+|x_3-x_4|+|x_4-x_1|).$$

通过定理 1 可知,当 $\lambda>(4\cdot1)/(2\cdot2)=1$ 时,$\tilde{f}_\lambda(\boldsymbol{x})$ 和 $f(x)$ 有着相同的最小值,并且 $\tilde{f}_\lambda(\boldsymbol{x})$ 的最小值点集合为 $\tilde{\chi}^*=\{x^*\mathbf{1}\mid x^*\in[2,4]\}$.

对于任意 $\lambda\leqslant1$,令 $\boldsymbol{x}=[2,2,4,4]^{\mathrm{T}}\notin\chi^*$.显然,

$$\tilde{f}_\lambda(\boldsymbol{x})=4+4\lambda\leqslant f^*=8,$$

这说明惩罚问题的最优值点集合不是 $\tilde{\chi}^*$.因此,对于任意 $\lambda\leqslant1$,我们不能通过求解惩罚问题 $\tilde{f}_\lambda(\boldsymbol{x})$ 来获得原问题 $f(x)$ 的解,说明在这个问题中定理 1 得到的下界(9)是紧的.

式(9)中的下界能否进一步减小呢?如能否用图的最小度来替换 $a_{\min}^{(l)}$,即 $d_m=\min_{i\in\mathcal{V}}\sum_{j=1}^{n}a_{ij}$?这在有些情况下是不可行的.考虑如图 1c 所示的所有边都是单位权重的 1-连通图.此时 $a_{\min}^{(1)}=1$ 以及 $d_m=2$.令 $[s_1,\cdots,s_6]=[1,2,3,4,5,6]$,$f_i(x)=|x-s_i|$,$\forall i\in\{1,\cdots,6\}$ 以及

$$\boldsymbol{x}=[x_1,\cdots,x_6]^{\mathrm{T}}=[3,3,3,4,4,4]^{\mathrm{T}}.$$

那么,经过和例 1 中类似的分析后,我们可以得出式(9)给出的下界 $\underline{\lambda}$ 不能更改为 $nc/(2d_m)=3/2$.

文献[14]针对使用常数步长的 DGD(3)给出了一个相似的惩罚函数法解释,其中选取的惩罚函数为

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{x}=\frac{1}{2}\sum_{i,j}a_{ij}(x_i-x_j)^2,$$

这里的 $\boldsymbol{L}$ 是图的拉普拉斯矩阵.然而,这样的二次型惩罚函数不能保证存在一个有限的 λ 使得原问题和惩罚问题等价.下面这个例子就这种情况.

例 2 考虑如图 1a 所示的所有边的权重都是 1 的一个无向图.令 $f_1(x)=(x-1)^2$ 以及 $f_2(x)=(x-3)^2$.显然,$f(x)=f_1(x)+f_2(x)$ 的最优值是 $x^*=2$.当选择惩罚函数为 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{x}$ 时,对应的惩罚问题是

$$\underset{x_1,x_2\in\mathbf{R}}{\text{minimize}}\ f_1(x_1)+f_2(x_2)+\lambda(x_1-x_2)^2. \quad (10)$$

不难看出,问题(10)的最优值是 $x_1^*=(1+4\lambda)/(1+2\lambda)$ 和 $x_2^*=(3+4\lambda)/(1+2\lambda)$.因此,不存在有限的 λ 使得这两个值与 $x^*=2$ 相等.

由文献[19]可知,$\boldsymbol{x}^*$ 是惩罚问题(6)的最优值点当且仅当 $0\in\partial\tilde{f}_\lambda(\boldsymbol{x}^*)$.定理 1 的(b)部分说明了对于任意 $\boldsymbol{x}\notin\{\alpha\mathbf{1}\mid\alpha\in\mathbf{R}\}$,惩罚问题在该点的次梯度的范数都会大于一个正的下界,从而 $\boldsymbol{x}$ 不是一个最优值点.

3 算法 1 在静态图上的收敛性

这一节研究算法 1 在静态图上的收敛性.当选择衰减的步长 ρ^k 时,我们证明了所有智能体的状态都会收敛到原问题(1)的同一个最优值点.此外,当步长为常数时,所有智能体的状态都收敛到最优值点的一个邻域内,并且邻域的大小与步长的大小成正比.我们还针对这两种步长分别给出了算法的收敛速度.

我们先介绍一个关于序列收敛性的引理.引理的证明可以参考文献[19]的 Proposition A.4.6.

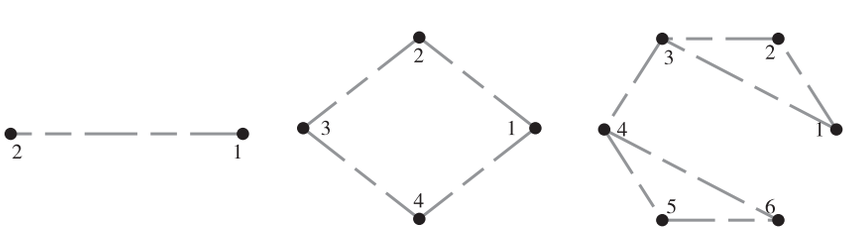
引理 1[19] 令 χ^* 表示 $\mathbf{R}^n$ 上的一个非空集,$\{\boldsymbol{x}^k\}\in\mathbf{R}^n$ 是满足下列条件的数列:

$$\|x^{k+1}-x^*\|^p\leqslant\|x^k-x^*\|^p-\gamma^k\varphi(x^k)+\delta^k,\quad\forall\boldsymbol{x}^*\in\chi^*,p>0,k>0,$$

其中 $\{\gamma^k\}$ 和 $\{\delta^k\}$ 都是非负的,并且

$$\sum_{k=0}^{\infty}\gamma^k=\infty,\ \sum_{k=0}^{\infty}\delta^k<\infty.$$

假设 $\phi(\cdot)$ 是一个连续、非负的函数,并且满足 $\phi(x)=0$ 当且仅当 $x\in\chi^*$.那么,$\{\boldsymbol{x}^k\}$ 收敛到 χ^* 中的某个点.



a. 无向图1 b. 无向图2 c. 无向图3

图 1 无向图示例[21]

Fig. 1 Some undirected graphs[21]

接下来证明算法 1 在衰减步长下的收敛性.为此,需要以下假设:

假设 2 步长序列$\{\rho^k\}$满足

$$\sum_{k=0}^{\infty}\rho^k=\infty,\quad \sum_{k=0}^{\infty}(\rho^k)^2<\infty.$$

在假设2成立的情况下,算法1的收敛性由下列定理给出.

定理 2(算法 1 的收敛性,文献[20]) 假设定理1中的条件和假设2成立.令$\{\boldsymbol{x}^k\}$表示算法1生成的数列.那么,存在某个$x^*\in\mathcal{X}^*$使得$\lim\limits_{k\to\infty}\boldsymbol{x}^k=x^*\mathbf{1}$.

当步长选择为$\rho^k=k^{-\alpha},\alpha\in(0.5,1]$时,我们可以刻画算法的收敛速度.为此,定义

$$d(\boldsymbol{x})=\min_{x^*\in\mathcal{X}^*}\|\boldsymbol{x}-x^*\mathbf{1}\|,$$

$$v(\boldsymbol{x}^k)=\max_i(x_i^k)-\min_i(x_i^k),$$

$$\bar{x}^k=\frac{1}{n}\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^k.$$

$d(\boldsymbol{x})$表示$\boldsymbol{x}$与最优值点集合之间的距离,v^k衡量了在k时刻不同智能体状态间的最大差异,以及$\bar{x}^k$表示在k时刻所有智能体状态的平均值.直观上,可以通过$f(\bar{x}^k)$趋近于f^*的速度以及v^k减小到0的速度来刻画算法1的收敛速度.

定理 3 假设定理1中的条件成立.令$\{\boldsymbol{x}^k\}$表示算法1生成的数列.令$\rho^k=k^{-\alpha},\alpha\in(0.5,1]$,那么,

$$\min_{1<k\leqslant\bar{k}}f(\bar{x}^k)-f^*\leqslant\frac{(2\alpha-1)d(\boldsymbol{x}^0)^2+2\alpha c_a^2}{2(2\alpha-1)s(\bar{k})},$$

$$\min_{1<k\leqslant\bar{k}}v(\boldsymbol{x}^k)\leqslant\frac{(2\alpha-1)d(\boldsymbol{x}^0)^2+2\alpha c_a^2}{(2\lambda a_{\min}^{(l)}-cn)(2\alpha-1)s(\bar{k})},\quad(11)$$

其中$\boldsymbol{x}^0$是初始状态,$\bar{x}^k$和$v(\boldsymbol{x}^k)$由(7)给出,以及

$$s(\bar{k})=\begin{cases}\dfrac{1}{1-\alpha}(\bar{k}^{1-\alpha}-1), & \alpha\in(0.5,1),\\ \ln(\bar{k}), & \alpha=1.\end{cases}$$

式(11)中第一个不等式刻画了$f(\bar{x}^k)$减小到f^*的速度,第二个不等式说明了智能体状态之间的最大差异以一个相近的速度衰减到0.因此,定理3表明算法1的收敛速度介于$O(1/\ln(k))$和$O(\ln(k)/\sqrt{k})$之间,具体取决于步长ρ^k的选择.

我们还给出了收敛速度的另一种刻画:

推论 1(算法 1 的收敛速度)[20] 假设定理3中的条件成立.那么

$$\min_{1<k\leqslant\bar{k}}\max_{i\in\mathcal{V}}f(x_i^k)-f^*\leqslant\frac{(2\alpha-1)d(\boldsymbol{x}^0)^2+2\alpha c_a^2}{2(2\alpha-1)s(\bar{k})},$$

其中所有符号的含义和定理3中一致.

当$f(x)$不可微时,经典的分布式梯度下降法(3)在步长选择为$\rho^k=1/\sqrt{k}$时以$O(\ln(k)/\sqrt{k})$的速度收敛.当α接近0.5时,这和算法1的收敛速度相当.因此,只使用相对状态的符号信息并没有在本质上降低算法的收敛速度.然而,当$f(x)$更光滑时,比如具有可微性或强凸性时,算法1可能要比(3)收敛得慢.

当选取常数步长时,算法1生成的$\boldsymbol{x}^k$以$O(1/k)$的速度接近最优值点集合的一个邻域,并且这个邻域的大小和步长的大小成正比.定理4和定理5说明了这个结论.

定理 4(常数步长的收敛性)[20] 假设定理1中的条件成立,并令$\{\boldsymbol{x}^k\}$表示算法1生成的数列.如果选择$\rho^k=\rho$,那么

$$\limsup_{k\to\infty}d(\boldsymbol{x}^k)\leqslant2\sqrt{n}\max\left\{\tilde{d}(\rho),\frac{\rho c_a^2}{2\lambda a_{\min}^{(l)}-cn}\right\}+\rho c_a,$$

其中$\tilde{\mathcal{X}}(\rho)=\{x\mid f(x)\leqslant f^*+\rho c_a^2/2\}$并且$\tilde{d}(\rho)=\max_{x\in\tilde{\mathcal{X}}(\rho)}d(x)<\infty$.

在定理4中,$\tilde{d}(0)=0$和$\tilde{d}(\rho)$关于ρ递增.因此,当选取常数步长时,算法1生成的状态最终会保持在$x^*\mathbf{1}$的一个邻域内,这里x^*可以是$\mathcal{X}^*$里的任意元素.这个邻域的大小随着ρ的减小而减小.如果f在最优值点附近的变化速率已知,那么$\tilde{d}(\rho)$甚至可以显式地表达出来.推论2给出了这个结果.

推论 2[20] 假设定理4的条件成立,并且$f(x)$满足

$$f(x)-f^*\geqslant\gamma(d(x))^{\alpha},$$

其中$\gamma>0$以及$\alpha\geqslant1$.那么有

$$\limsup_{k\to\infty}d(\boldsymbol{x}^k)\leqslant2\sqrt{n}\max\left\{\left(\frac{\rho c_a^2}{2\gamma}\right)^{\frac{1}{\alpha}},\frac{\rho c_a^2}{2\lambda a_{\min}^{(l)}-cn}\right\}+\rho c_a.$$

下面定理给出了常数步长下算法的收敛速度.

定理 5[20] 假设定理4中的条件成立.那么

$$\min_{0\leqslant k\leqslant\bar{k}}f(\bar{x}^k)-f^*\leqslant\frac{\rho c_a^2}{2}+\frac{d(\boldsymbol{x}^0)^2}{2\rho\bar{k}},$$

$$\min_{0\leqslant k\leqslant\bar{k}}v(\boldsymbol{x}^k)\leqslant\frac{\rho c_a^2}{2\lambda a_{\min}^{(l)}-cn}+\frac{d(\boldsymbol{x}^0)^2}{\rho\bar{k}(2\lambda a_{\min}^{(l)}-cn)}.\quad(12)$$

注 3 根据定理5易得以下结论:

(a) $\min\limits_{0\leqslant k\leqslant\bar{k}}f(\bar{x}^k)$以$O(1/\bar{k})$的速度趋近区间$\left[f^*,f^*+\dfrac{\rho c_a^2}{2}\right]$.

(b) 当$\bar{k}$固定时,选择ρ^k使得式(12)中不等式

右边部分最小,即 $\rho^k=\frac{1}{c_a}\frac{d(\boldsymbol{x}^0)}{\sqrt{k}}$.那么

$$\min_{0\leqslant k\leqslant k} f(\bar{x}^k)-f^*\leqslant c_a\frac{d(\boldsymbol{x}^0)}{\sqrt{k}},$$

$$\min_{0\leqslant k\leqslant k} v(\boldsymbol{x}^k)\leqslant \frac{c_a}{2\lambda a_{\min}^{(l)}-cn}\frac{d(\boldsymbol{x}^0)}{\sqrt{k}},$$

也就是说,智能体的状态只能收敛到距离某个最优值点大小为 $O(\bar{k}^{-1/2})$ 的邻域内.

4 时变图上的分布式算法

当多智能体系统之间的通信拓扑是时变时,算法 1 需要做一些修正,参见算法 2 . 这一节分析算法 2 在两种不同类型的时变图上的表现,分别是一致连通图和随机激活图.

4.1 一致连通图

首先介绍一致连通图的概念. 令 $\mathcal{G}^{(k,b)}$ 表示从第 k 时刻开始连续 b 个时刻以内的图的并,即对于所有正整数 k 以及正数 b,有 $\mathcal{G}^{(k,b)}=(\mathcal{V},\mathcal{E}^{(k,b)},\boldsymbol{A}^{(k,b)}):=\mathcal{G}^k\cup\mathcal{G}^{k+1}\cup\cdots\cup\mathcal{G}^{k+b-1}$,其中 $\boldsymbol{A}^{(k,b)}$ 表示图 $\mathcal{G}^{(k,b)}$ 对应的邻接矩阵.

假设 3 假设:

(a) 存在 $\eta>0$ 使得

$$\begin{cases}a_{ij}^k\geqslant\eta, & \text{如果}(i,j)\in\mathcal{E}^k,\\ a_{ij}^k=0, & \text{其他.}\end{cases}$$

(b) 存在正整数 $b\geqslant 1$ 使得对于任意 $t=0,1,2,\cdots$,邻接矩阵 $\boldsymbol{A}^{(tb,b)}$ 对应的图为 l- 连通图.

假设 3 在处理确定性时变图时很常见.其中第一部分要求图中的任意一条边在某个时刻要么不连通,要么连通后的权重大于某个正的下界.第二部分假设的是任意连续 b 时刻内图的并是连通的. 我们称满足假设 3 的时变图为一致连通图,在有些文献中也被叫做 b- 连通图[4,8].

下面给出算法 2 在一致连通图上的收敛性结论.

定理 6(算法 2 在一致连通图上的收敛性)[22] 假设 1—3 成立,并且存在常数 $c_\rho>0$ 使得对于任意 $k>0$,

$$\max_{t\in[k,k+b)}\rho^t\leqslant c_\rho\min_{t\in[k,k+b)}\rho^t.$$

如果选择

$$\lambda>\frac{nbcc_\rho}{2l\eta},$$

其中 n 表示智能体的个数,c 由假设 1 给出,c_ρ 由假设 2 给出,b,l,η 由假设 3 给出.令 $\{\boldsymbol{x}^k\}$ 表示算法 2 生成的数列.那么,存在 $x^*\in\mathcal{X}^*$,使得 $\lim_{k\to\infty}\boldsymbol{x}^k=x^*\boldsymbol{1}$.

与静态图上的收敛性结论相比(定理 2),一致连通图上的收敛性结论的最大不同在于 λ 的选择与静态图中相比需要增大 bc_ρ 倍.

接下来给出当步长选择为 $\rho^k=k^{-\alpha},\alpha\in(0.5,1]$ 时算法 2 在一致连通图上的收敛速度.与定理 3 类似,我们通过给出 $f(\bar{x}^k)$ 趋近 f^* 的速度以及 $v(\boldsymbol{x}^k)$ 减小到 0 的速度来衡量算法的收敛速度.

定理 7(算法 2 在一致连通图上的收敛速度[22]) 假设定理 6 中的条件成立以及 $\lambda>nbc/l\eta$. 令 $\{\boldsymbol{x}^k\}$ 表示算法 2 生成的数列. 如果 $\rho^k=k^{-\alpha}$,其中 $\alpha\in(0.5,1]$,那么对于任意 $k_0>2b$,有

$$\min_{1<k\leqslant k_0} f(\bar{x}^k)-f^*\leqslant\frac{(2\alpha-1)(d(\boldsymbol{x}^0))^2+10\alpha bc_a^2}{b(2\alpha-1)s(k_0)},$$

$$\min_{1<k\leqslant k_0} v(\boldsymbol{x}^k)\leqslant\frac{2(2\alpha-1)(d(\boldsymbol{x}^0))^2+12\alpha bc_a^2}{(\lambda l\eta-nbc)(2\alpha-1)s(k_0)},$$

其中 $\boldsymbol{x}^0$ 表示初始状态,

$$s(k_0)=\begin{cases}\dfrac{(k_0-b)^{1-\alpha}-b^{1-\alpha}}{b(1-\alpha)}, & \alpha\in(0.5,1),\\ \dfrac{1}{b}[\ln(k_0-b)-\ln(b)], & \alpha=1.\end{cases}$$

定理 2 说明,在最坏的情况下,算法在一致连通图上的收敛速度要比在静态图上的收敛速度(定理 3)慢 b 倍. 直观上看,这是合理的.

算法 2:基于相对状态符号信息的分布式优化算法(时变图)

1) 初始化.对于每个节点 $i\in\mathcal{V}$,令 $x_i^0=0$.

2) 开始循环.

3) 信息交换.每个节点 $i\in\mathcal{V}$ 获得其与当前时刻的邻居节点 $j\in\mathcal{N}_i^k$ 之间的相对状态信息的符号,并按照如下公式计算 r_i^k:

$$r_i^k=\sum_{j\in\mathcal{N}_i^k}a_{ij}^k\operatorname{sgn}(x_j^k-x_i^k).$$

4) 状态更新.按照如下方式更新每个节点 $i\in\mathcal{V}$ 的自身状态:

$$x_i^{k+1}=x_i^k+\rho^k(\lambda\cdot r_i^k-\nabla f_i(x_i^k)),$$

其中 λ 和 ρ^k 由定理 6 给出,$\nabla f_i(x_i^k)$ 为 $f_i(x)$ 在 x_i^k 处的任意次梯度.

5) 令 $k=k+1$.

6) 终止条件.如果一个预先设定的终止条件(如最大迭代次数)得到满足,则算法终止.

4.2 随机激活图

这一小节给出算法 2 在随机激活图上的收敛性结论.随机激活图可以用来描述包括 gossip 社交网络以及通信网络中的丢包在内的一些实际应用.其定义如下:

定义 2(随机激活图) 令$\{\mathcal{G}^k\}$表示不同时刻的图构成的序列.如果对于任意$i,j\in\mathcal{V},i\neq j,\{a_{ij}^k\}$都是一个独立同分布的伯努利过程,且对应参数为$\mathbb{P}\{a_{ij}^k=1\}=p_{ij}$,那么,我们称这种时变图为随机激活图.上式中$\mathbb{P}(\mathcal{X})$表示事件$\mathcal{X}$发生的概率,且$0\leqslant p_{ij}\leqslant 1,\ \forall i,j\in\mathcal{V}$.

注 4 为了简化符号,在随机激活图中我们假设任意时刻每条边的权重a_{ij}^k为 0 或 1.

我们把$\boldsymbol{P}=[p_{ij}]$叫做图$\mathcal{G}^k$的激活概率矩阵,并且把$\boldsymbol{P}$对应的图表示成$\mathcal{G}_P$.$\mathcal{G}_P$可以看作是$\mathcal{G}^k$的期望图,即,

$$\mathcal{G}_P:=\mathbb{E}(\mathcal{G}^k).$$

如前所述,算法 1 可以看作是惩罚问题(6)的次梯度方法.类似地,算法 2 也可以看作是下面优化问题的随机次梯度方法:

$$\underset{\boldsymbol{x}\in\mathbf{R}^n}{\text{minimize}}\ \hat{f}_\lambda(\boldsymbol{x}):=g(\boldsymbol{x})+\lambda\hat{h}(\boldsymbol{x}),$$

其中$g(\boldsymbol{x})$在(4)中给出.

$$\hat{h}(\boldsymbol{x})=\frac{1}{2}\sum_{i,j=1}^{n}p_{ij}|x_i-x_j|.$$

这个结果不难证明.注意到$\mathbb{E}(a_{ij}^k)=p_{ij}$,因此$\hat{h}(\boldsymbol{x})$的一个次梯度$\nabla_s\hat{h}(\boldsymbol{x})=[\nabla_s\hat{h}(\boldsymbol{x})_1,\cdots,\nabla_s\hat{h}(\boldsymbol{x})_n]^{\mathrm{T}}$可以按照元素写成

$$\nabla_s\hat{h}(\boldsymbol{x})_i=\sum_{j=1}^{n}a_{ij}^k\operatorname{sgn}(x_i-x_j)=\sum_{j\in\mathcal{N}_i^k}\operatorname{sgn}(x_i-x_j).$$

因为$\mathbb{E}\{\nabla_s\hat{h}(\boldsymbol{x})_i\}=\sum_j p_{ij}\operatorname{sgn}(x_i-x_j)$,所以$\mathbb{E}\{\nabla_s\hat{h}(\boldsymbol{x})\}$是$\hat{h}(\boldsymbol{x})$的一个次梯度.下面引理给出了随机次梯度算法的收敛性,利用它可以证明算法 2 会依概率 1 收敛到原问题的一个最优值点.

引理 2(随机梯度算法的收敛性[23]) 考虑如下优化问题:

$$\underset{\boldsymbol{x}\in\mathbf{R}^n}{\text{minimize}}\ \mathbb{E}\{F(\boldsymbol{x},w)\},\tag{14}$$

其中w是随机变量并且对于任意w,$F(\boldsymbol{x},w):\mathbf{R}^n\times\mathbf{R}\to\mathbf{R}$是关于$\boldsymbol{x}$的连续凸函数.令$\mathcal{X}^*$表示该问题的最优值点集合并假设$\mathcal{X}^*$非空.

下式给出了问题(14)的随机次梯度算法:

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\rho^k r(\boldsymbol{x}^k,w^k),$$

其中$r(\boldsymbol{x},w^k)$是一个有界的随机向量并且满足对于所有$\boldsymbol{x}\in\mathbf{R}^n$,$\mathcal{E}(r(\boldsymbol{x},w^k))$是$\mathcal{E}\{F(\boldsymbol{x},w^k)\}$的一个次梯度.如果$\{\rho^k\}$满足

$$\sum_{k=0}^{\infty}\rho^k=\infty,\quad\sum_{k=0}^{\infty}(\rho^k)^2<\infty,$$

那么,存在$\boldsymbol{x}^*\in\mathcal{X}^*$使得

$\lim\limits_{k\to\infty}\boldsymbol{x}^k=\boldsymbol{x}^*$依概率 1 成立.

下面定理综合了以上分析,是本小节的主要结论.

定理 8(算法 2 在随机激活图上的收敛性[20]) 假设 1 和 2 成立,并且$\mathcal{G}_P$对应的图是l-连通图.如果

$$\lambda>\frac{nc}{2p_{\min}^{(l)}},$$

其中$\mathcal{G}_P$由(13)给出,$p_{\min}^{(l)}$表示$\boldsymbol{P}$中最小的l个非零元素之和,那么,令$\{\boldsymbol{x}^k\}$表示算法 2 生成的数列,则对于某个$\boldsymbol{x}^*\in\mathcal{X}^*$,$\lim\limits_{k\to\infty}\boldsymbol{x}^k=\boldsymbol{x}^*\mathbf{1}$依概率 1 成立.

5 数值实验

本节利用所提算法分布式地寻找平面上若干个点的几何中心,所有实验结果与文献[21]一致.平面上n个点的几何中心是指平面上到这n个点的欧式距离和最小的点[24].另一方面,几何中心也是以下优化问题的解:

$$\underset{\boldsymbol{x}\in\mathbf{R}^2}{\text{minimize}}\ f(\boldsymbol{x}):=\sum_{i=1}^{n}f_i(\boldsymbol{x})=\sum_{i=1}^{n}\|\boldsymbol{x}-\boldsymbol{x}_i\|,\tag{15}$$

其中$\boldsymbol{x}_i(i=1,\cdots,n)$为$n$个点的坐标,$f_i(\boldsymbol{x}):=|\boldsymbol{x}-\boldsymbol{x}_i|$是一个凸函数,但是不可微,其次微分由下式给出:

$$\partial f_i(\boldsymbol{x})=\begin{cases}\dfrac{\boldsymbol{x}-\boldsymbol{x}_i}{\|\boldsymbol{x}-\boldsymbol{x}_i\|}, & \text{如果}\ \boldsymbol{x}\neq\boldsymbol{x}_i,\\ \{g\mid\|g\|\leqslant 1\}, & \text{其他.}\end{cases}$$

显然,问题(15)满足假设 1,可以通过算法 1 和算法 2 进行求解.注意式(15)中的$\boldsymbol{x}$是一个二维向量,因此算法 1 和算法 2 需要按照注 1 做一些修正.

上述的几何问题可以看作是统计学中的最小二乘问题和区位理论中的 Weber 问题的一个特例.这里我们给出其可能的一种分布式应用场景:假设有n个基站分布在海底,而我们希望找到一个位置来建立通信中心,使得通信中心到所有基站的距离和最短以节约线缆成本.因为在海底无法使用全球定位系统,即无法获取基站的绝对位置,所以一个可行的办法是分布式地寻找基站的几何中心.具体来说,利用本文提出的算法,每个基站派出一个智能体,这个智能体只需要测量它与该基站的距离,并且只需要大概知道它的邻居的方位就可以找到这个几何

中心.

在接下来的仿真实验中,我们考虑 5 个基站的情形(即有 5 个智能体),且基站随机地分布在一个 100 × 100 的正方形区域内.我们分别在静态图、一致连通图以及随机激活图上进行了仿真.在所有仿真中,选取步长为$\rho^k = 5/(k+10)$.5 个智能体之间的通信拓扑是个环形网络,如图 2a 所示.在静态图仿真中,选取算法 1 中的$\lambda = 2$,使其满足注 2 中的条件.在一致连通图中,我们在每一时刻只让一条边连通,并且 5 条边依次连通.在这种情况下选取算法 2 中的$\lambda = 6$.在随机激活图中,令每条边在每一时刻连通的概率为 0.5,并且选择$\lambda = 4$.

图 2b、2c、2d 分别画出了 5 个智能体在静态图、一致连通图和随机激活图上的仿真轨迹,其中实心圆表示智能体的初始位置,黑色三角形表示由 Weiszfeld 方法[25]计算得到的这些实心圆的几何中心.如图所示,在 3 种情况中智能体都收敛到了几何中心,不过收敛过程有所不同.

如果选取比定理 1 中给出的下界更小的λ,智能体之间可能不能趋于一致.图 3 画出了当分别在静态图、一致连通图以及随机激活图上选取$\lambda = 0.8, 2, 1.5$ 时智能体的轨迹.除了将仿真步数增加到了 5 000,仿真的其他设置和之前保持一致.显然,在这 3 种情况下智能体都不能趋同.

6 总结

本文讨论了一种分布式优化算法求解多智能体系统中具有加和形式目标函数的优化问题,该算法的特点是只需要利用节点之间相对状态的符号信息.此外,该算法可以应用于静态以及时变的网络结构.在静态的网络结构下,首先利用优化理论中的惩罚函数方法来解释了所提算法,然后分析了算法在

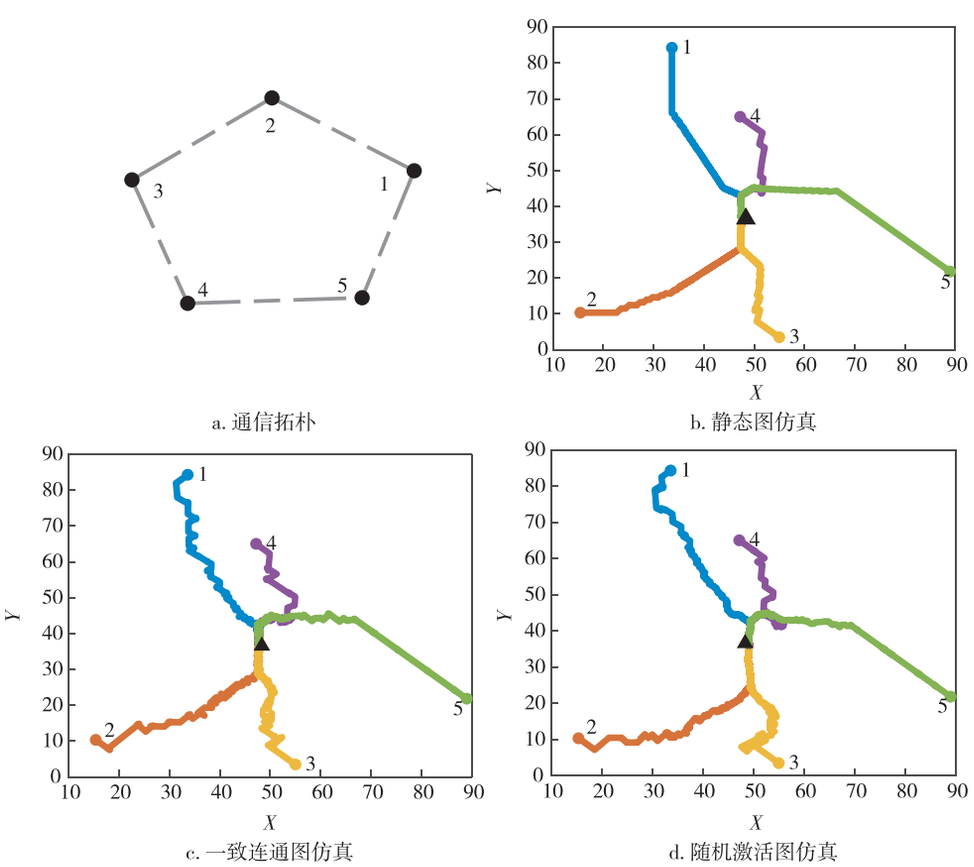
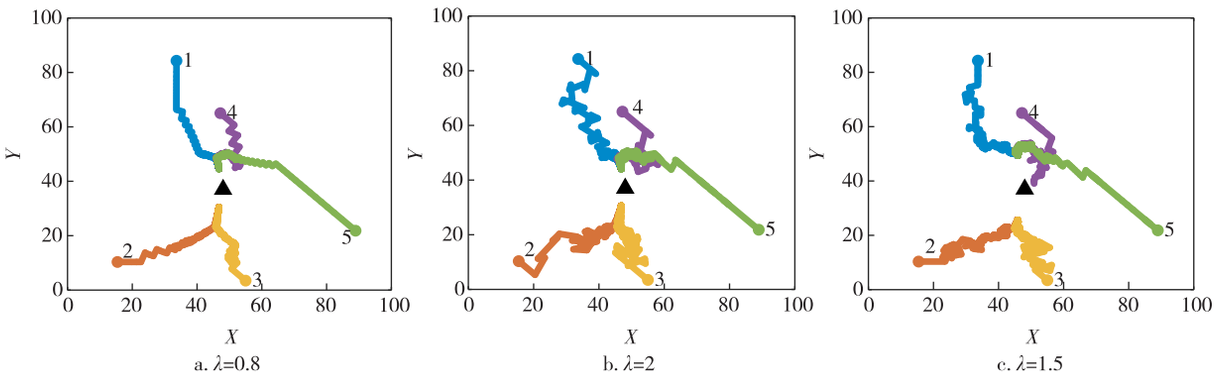


图 2[21] (a)智能体之间的通信拓扑,(b) 静态图仿真中智能体的轨迹(其中实心圆表示智能体的初始位置,黑色三角形表示这些实心圆对应的几何中心),(c) 一致连通图仿真中智能体的轨迹,(d) 随机激活图仿真中智能体的轨迹

Fig. 2[21] (a)The topology of the agents,(b) The trajectories of the agents in a static graph, where the filled circles are the initial positions of the agents and the black triangle is the geometric median of these circles,(c) The trajectories of the agents in uniformly jointly connected graphs,(d) The trajectories of the agents in randomly activated graphs



a. λ=0.8　b. λ=2　c. λ=1.5

图 3　当选取更小一些的 λ 时智能体在静态图、一致连通图以及随机激活图上的轨迹[21]

Fig. 3　The trajectories of agents with smaller λ over a static graph, uniformly jointly connected graphs, and randomly activate graphs, respectively[21]

衰减步长和常数步长下的收敛性,最后证明了算法的收敛速度介于 $O(1/\ln(k))$ 和 $O(1/\sqrt{k})$ 之间.在时变网络下,我们讨论了算法在一致连通图和随机激活图下的性能,并给出了相应的收敛性结论.最后,通过一个分布式寻找几何中心的例子,利用数值仿真实验说明了算法的有效性,并进一步验证了理论结果.

致谢　本文作者非常感谢 Tamer Başar 教授对本文内容提出的一些宝贵建议.本文工作受到了国家自然科学基金(61722308)以及国家重点基础研究专项基金(2017YFC0805310)的资助.

参考文献

References

[1] You K,Xie L.Network topology and communication data rate for consensusability of discrete-time multi-agent systems[J].IEEE Transactions on Automatic Control,2011,56(10):2262-2275

[2] You K,Tempo R,Xie P.Distributed algorithms for robust convex optimization via the scenario approach[J].IEEE Transactions on Automatic Control,2018,DOI:10.119/TAC.2018.2828093

[3] Cevher V,Becker S,Schmidt M.Convex optimization for big data:scalable,randomized,and parallel algorithms for big data analytics[J].IEEE Signal Processing Magazine,2014,31(5):32-43

[4] Nedić A,Olshevsky A,Rabbat M G.Network topology and communication-computation tradeoffs in decentralized optimization[J].Proceedings of the IEEE,2018,106(5):953-976

[5] Wang H,Li C.Distributed quantile regression over sensor networks[J].IEEE Transactions on Signal & Information Processing Over Networks,2017,DOI:10.1109/TSIPN.2017.2699923

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

[7] Li T,Fu M,Xie L,et al.Distributed consensus with limited communication data rate[J].IEEE Transactions on Automatic Control,2011,56(2):279-292

[8] Nedic A,Olshevsky A.Distributed optimization over time-varying directed graphs[C]// Decision and Control.IEEE,2013:6855-6860

[9] Olfati-Saber R,Murray R M.Consensus problems in networks of agents with switching topology and time-delays[J].IEEE Transactions on Automatic Control,2004,49(9):1520-1533

[10] Yi P,Hong Y.Quantizedsubgradient algorithm and data-rate analysis for distributed optimization[J].IEEE Transactions on Control of Network Systems,2014,1(4):380-392

[11] Pu Y,Zeilinger M N,Jones C N.Quantization design for distributed optimization[J].IEEE Transactions on Automatic Control,2017,62(5):2107-2120

[12] Magnússon S,Enyioha C,Li N,et al.Convergence of limited communications gradient methods[J].IEEE Transactions on Automatic Control,2017,DOI:10.1109/TAC.2017,2743678

[13] Shi W,Ling Q,Wu G,et al.EXTRA:an exact first-order algorithm for decentralized consensus optimization[J].SIAM Journal on Optimization,2014,25(2):944-966

[14] Mokhtari A,Ling Q,Ribeiro A.Network newton distributed optimization methods[J].IEEE Transactions on Signal Processing,2017,65(1):146-161

[15] Chen G,Lewis F L,Xie L.Finite-time distributed consensus via binary control protocols[J].Automatica,2011,47(9):1962-1968

[16] Franceschelli M,Giua A,Pisano A.Finite-time consensus on the median value with robustness properties[J].IEEE Transactions on Automatic Control,2017,62(4):1652-1667

[17] Lin P,Ren W,Farrell J A.Distributed continuous-time

optimization: nonuniform gradient gains, finite-time convergence, and convex constraint set [J]. IEEE Transactions on Automatic Control, 2017, 62(5): 2239-2253
[18] Clarke F H, Stern R J, Ledyaev Y S, et al. Nonsmooth analysis and control theory[J]. Graduate Texts in Mathematics, 1998, 178(7): 137-151
[19] Bertsekas D P. Convex optimization algorithms [M]. Athena Scientific, 2016
[20] Zhang J, You K, Başar T. Distributed discrete-time optimization in multi-agent networks using only sign of relative state [J]. IEEE Transactions on Automatic Control, 2018, Conditionally Accepted
[21] ZHANG Jiaqi, YOU Keyou. Distributed optimization in multi-agent networks using one-bit of relative state information[M]// Basar Tamer. Uncertainty in Complex Networked Systems, Birkäuser, 2018
[22] Zhang J, You K. Distributed optimization with binary relative information over deterministically time-varying graphs[C]// The 57th IEEE Conference on Decision and Control, Miami Beach, FL, USA, 2018, Accepted
[23] Borkar V S. Stochastic approximation: a dynamical systems viewpoint[C]// Baptism's 91 Witnesses, 2008
[24] CohenM B, Lee Y T, Miller G, et al. Geometric median in nearly linear time[C]// Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, ACM, 2016: 9-21
[25] Beck A, Sabach S. Weiszfeld's method: old and new results[J]. Journal of Optimization Theory and Applications, 2015, 164(1): 1-40

Distributed optimization using the sign of relative state information

ZHANG Jiaqi[1,2] YOU Keyou[1,2]
1 Department of Automation, Tsinghua University, Beijing 100084
2 Beijing National Research Centre for Information Science and Technology, Tsinghua Univesity, Beijing 100084

Abstract The design of distributed discrete-time algorithms to cooperatively solve an additive cost optimization problem in multi-agent networks is presented in this paper. The striking feature of the distributed algorithms lies in the use of only the sign of the relative state information between neighbors; which substantially differentiates our algorithms from the existing ones. Moreover, the algorithm does not require the interaction matrix to be doubly stochastic. We first interpret the proposed algorithms in terms of the penalty method in the optimization theory and then perform a non-asymptotic analysis to study the convergence for static network graphs. Compared with the celebrated distributed subgradient algorithms, which, however, use the exact relative state information, the convergence speed in the proposed algorithms is essentially not affected by the loss of information. We also extend our results to the cases of deterministically and randomly time-varying graphs. Finally, we validate the theoretical results through simulations.

Key words distributed optimization; multi-agent networks; sign of relative state; penalty method; subgradient iterations