Automatica 94 (2018) 55–62

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Augmented Lagrange algorithms for distributed optimization over multi-agent networks via edge-based method☆

Chong-Xiao Shi, Guang-Hong Yang *

College of Information Science and Engineering, Northeastern University, Shenyang 110819, PR China
State Key Laboratory of Synthetical Automation of Process Industries, Northeastern University, Shenyang, 110819, PR China

ARTICLE INFO

Article history:
Received 29 July 2017
Received in revised form 6 December 2017
Accepted 13 March 2018

Keywords:
Distributed optimization
Augmented Lagrange method
Convergence analysis
Factorization of weighted Laplacian

ABSTRACT

In this paper, the augmented Lagrange (AL) algorithm for distributed optimization is studied. Compared with the existing results, this paper uses different techniques, including the factorization of weighted Laplacian and the spectral decomposition technique, to prove the linear convergence of the AL algorithm, and simultaneously provides a novel description on the convergence rate. First, by using an important factorization of weighted Laplacian, it is proved that the linear convergence of the AL algorithm can be achieved via a simplified analysis procedure. Within this framework, a novel quantitative description on the convergence rate is then provided based on spectral decomposition technique. Meanwhile, by determining the monotonicity of an auxiliary function, a connection between convergence rate, step size and edge weights is established. Finally, simulation examples illustrate the theoretical results.

1. Introduction

In the past few years, the study on multi-agent networks has attracted much attention with the rapid development of networked systems (Li & Yang, 2016; Liu & Wang, 2013; Ma & Yang, 2016; Meng, Xiao, & Xie, 2011). Especially, distributed optimization over multi-agent networks is one of the most popular topics due to its widely applications on source localization, data regression, model predictive control and resource allocation (Guo, Wen, Mao, & Song, 2016; Mota, Xavier, Aguiar, & Püschel, 2015; Yi, Hong, & Liu, 2016; Zhang, Lou, Hong, & Xie, 2015). In distributed optimization, a network of agents is often considered to cooperatively minimize the sum of local objective functions, where the local objective function is privately known by its individual agent, and each agent can only exchange the information with its neighbors.

The existing distributed optimization algorithms are mostly constructed by using primal domain (gradient descent) methods and primal dual (augmented Lagrange) methods. In primal domain methods, Nedić and Ozdaglar (2009) present a distributed subgradient algorithm to solve the convex optimization problems over time-varying networks, and the subgradient algorithm is extended in Nedić, Ozdaglar, and Parrilo (2010) to solve the constrained optimization problems. The works in Nedić and Olshevsky (2015), Tsianos, Lawlor, and Rabbat (2012) and Xi and Khan (2016) generalize the distributed subgradient algorithms from the undirected graphs to the directed ones. A fast distributed algorithm is presented in Jakovetić, Xavier, and Moura (2014) by using Nesterov's gradient method, and a dual averaging subgradient method is presented in Duchi, Agarwal, and Wainwright (2012). Note that the advantage of primal domain methods is their low computation burden. However, this class of methods often leads to slow convergence or low accuracy due to the requirement of diminishing step sizes (Ling, Shi, Wu, & Ribeiro, 2015).

To accelerate the convergence, some efficient distributed algorithms with fixed step sizes (Qu & Li, 2017; Xu, Zhu, Soh, & Xie, 2015) are provided. Especially, the well known distributed augmented Lagrange (AL) methods often exhibit the fast linear convergence. One type of AL methods is named as Alternating Direction Method of Multipliers (ADMM), based on which many effective distributed algorithms are proposed, such as Chang, Hong, and Wang (2015), Iutzeler, Bianchi, Ciblat, and Hachem (2016), Makhdoumi and Ozdaglar (2017), Shi, Ling, Yuan, Wu, and Yin (2014) and Wei and Ozdaglar (2012a). However, distributed ADMM often leads to high computation burden since the primal variables need to achieve successive minimizations at each iteration (Ling et al., 2015). To reduce the computation burden, the linearized ADMM algorithms (Aybat, Wang, Lin, & Ma, 2018; Ling et al., 2015) as

☆ This work was supported in part by the Funds of the National Natural Science Foundation of China (Grant Nos. 61621004 and 61420106016), and the Research Fund of State Key Laboratory of Synthetical Automation for Process Industries (Grant No. 2013ZCX01). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Julien M. Hendrickx under the direction of Editor Christos G. Cassandras.

* Corresponding author at: College of Information Science and Engineering, Northeastern University, Shenyang 110819, PR China.
E-mail addresses: s254705600@sina.com (C.-X. Shi), yangguanghong@ise.neu.edu.cn (G.-H. Yang).

https://doi.org/10.1016/j.automatica.2018.04.010
0005-1098/

well as the saddle point methods are studied, where only one gradient-like step is performed at each iteration to update the primal variables, thus leads to lower computation burden. The exact first-order algorithm (EXTRA) (Shi, Ling, Wu, & Yin, 2015) is one of the classical saddle point methods, based on which some interesting generalized algorithms (Hong, 2016; Mokhtari & Ribeiro, 2016; Nedić, Olshevsky, & Shi, 2017) are developed. It has been shown in Ling et al. (2015) and Shi et al. (2015) that both EXTRA methods and linearized ADMM algorithms can also achieve the linear convergence, which enjoy the advantages of both lower computation burden and fast convergence. In fact, the convergence analyses of these algorithms are very crucial on establishing the linear convergence, and also closely relate to the rate description. Based on these observations, one of the most important problems is that how to simplify the convergence analyses of the existing methods and further provide a more concise description on convergence rate. Meanwhile, it is also of interest to establish a novel relationship between the convergence rate, step size and graph parameters. These reasons motivate our current study.

In this paper, the AL algorithm is further studied. The main contribution of this paper is that a simpler and clearer presentation of the convergence proof for the gradient-based AL algorithm is provided. Specifically, this paper uses different techniques, including the factorization of weighted Laplacian and the spectral decomposition technique, to prove the linear convergence of the algorithm. Compared with the existing results, the differences of our work are summarized as follows:

- It is shown that the convergence analysis simplifies the proofs of EXTRA methods (Mokhtari & Ribeiro, 2016; Shi et al., 2015) and linearized ADMM algorithms (Aybat et al., 2018; Ling et al., 2015). Within this framework, a novel description on the convergence rate is provided, which has a simpler relation associated with the communication graphs than the EXTRA methods (Mokhtari & Ribeiro, 2016; Shi et al., 2015) and the ADMM algorithms (Ling et al., 2015; Makhdoumi & Ozdaglar, 2017; Shi et al., 2014).
- Further, a connection between the convergence rate, step size and edge weights is established, which provides a different insight on the rate description compared with the methods over unweighted networks (Chang et al., 2015; Iutzeler et al., 2016; Shi et al., 2014; Wei & Ozdaglar, 2012a), as well as the gradient method (Qu & Li, 2017) that focuses on the relation between the convergence rate and function smoothness.
- Besides, the convergence of our algorithm is guaranteed when the step size is smaller than a constant. This is similar to the gradient based AL methods such as Kia, Cortés, and Martínez (2015), Lei, Chen, and Fang (2016), Liu, Yang, and Hong (2017), Shi et al. (2015) and Wang and Elia (2010), but differs from the ADMM algorithms such as Makhdoumi and Ozdaglar (2017) and Shi et al. (2014).

In addition, it is worth mentioning that the proposed algorithm needs to know a global ordering on the nodes, which is similar to the previous works (Mota, Xavier, Aguiar, & Püschel, 2013; Wei & Ozdaglar, 2012a). In fact, this is a necessary assumption in our algorithm design. One technique to relax this assumption might be changing the order of node update into a random way, as suggested in Wei and Ozdaglar (2012b, 2013c). However, using this technique in our framework may arise new challenges for the convergence analysis of algorithm. Thus how to relax this assumption still requires further studies.

Paper organization: Section 2 gives the problem formulation and the algorithm design. The convergence analysis and the convergence rate description are provided in Section 3. In Section 4, the simulation examples are given, and finally, the conclusions are derived in Section 5.

Notation: For any matrix $A \in \mathbb{R}^{n\times m}$, A^T represents its transpose. $\mathbf{1}_N$ denotes the $N\times 1$ vector with all elements equal to 1. I_N denotes the $N\times N$ identity matrix. $\otimes$ represents the Kronecker product. $diag(\bullet)$ denotes the diagonal matrix. For a vector $v \in \mathbb{R}^n$, $\|v\|_2$ denotes the Euclidian norm. For a given semidefinite matrix G with proper dimensions, G-norm of v is denoted as $\|v\|_G = \sqrt{v^T G v}$.

2. Problem formulation

2.1. Graph theory

An undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ consists of a set $\mathbb{V} = \{v_1, v_2, \ldots, v_N\}$ of N nodes, and a set $\mathbb{E}$ of M edges (v_j, v_i), where the order of the nodes in $\mathbb{E}$ is irrelevant. The neighborhood of ith node is denoted as $\mathcal{N}_i$. An undirected graph $\mathbb{G}$ is connected if every pair of nodes is joined by a path. The weight l_{ij} associated with each edge is a nonnegative real value, and if there is no edge connecting v_i and v_j, $l_{ij} = 0$. The Laplacian matrix $\mathcal{L} = [\mathcal{L}_{ij}] \in \mathbb{R}^{N\times N}$ for an undirected graph is defined as follows: $\mathcal{L}_{ij} = -l_{ij}$ if $i \neq j$; and $\mathcal{L}_{ij} = \sum_{j\neq i}^{N} l_{ij}$ otherwise. Without loss of generality, let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ be the eigenvalues of $\mathcal{L}$.

Next, we give a factorization of the Laplacian matrix $\mathcal{L}$. Denote weight matrix $\mathcal{W} = diag(w_1, w_2, \ldots, w_M)$ as an $M \times M$ diagonal matrix with diagonal elements given by the edge weights, i.e., $w_\iota = l_{ij}$, *for* $(i, j) = e_\iota, \iota = 1, 2, \ldots, M$. Subsequently, each edge of $\mathbb{G}$ is assigned with an arbitrary orientation, i.e., for each edge $e_\iota \in \mathcal{E}$, denote one endpoint as the head and the other as the tail. Then, we define the oriented incidence matrix $\mathcal{E} = [\varepsilon_{\kappa\iota}] \in \mathbb{R}^{N\times M}$ as follows: $\varepsilon_{\kappa\iota} = 1$ if node κ is the head of e_ι; $\varepsilon_{\kappa\iota} = -1$ if node κ is the tail of e_ι; and $\varepsilon_{\kappa\iota} = 0$ otherwise. It should be pointed out that the incident matrix $\mathcal{E}$ has an important property that $\mathbf{1}^T\mathcal{E} = 0$. Similar to Wei and Ozdaglar (2012a), it is assumed that the smaller row in each column of $\mathcal{E}$ has entry 1 while the larger one has -1. Further, based on the definitions of the weight matrix $\mathcal{W}$ and incidence matrix $\mathcal{E}$, the Laplacian matrix $\mathcal{L}$ can be factorized as $\mathcal{L} = \mathcal{E}\mathcal{W}\mathcal{E}^T = \mathcal{E}\mathcal{W}^{\frac{1}{2}}\mathcal{W}^{\frac{1}{2}}\mathcal{E}^T$. Note that this factorization plays an important role in the later developments.

2.2. Problem formulation

In this paper, a network of N agents is considered to cooperatively solve the following optimization problem:

$$\min_{\tilde{x}} \tilde{f}(\tilde{x}) = \sum_{i=1}^{N} f_i(\tilde{x}), \tag{1}$$

where $\tilde{x} \in \mathbb{R}^n$, $f_i(\tilde{x}) : \mathbb{R}^n \to \mathbb{R}$ is the convex objective function. It is assumed that the minimum of (1) can be attained. Moreover, the function $f_i(\tilde{x})$ is privately known by ith agent and each agent can only communicate with its neighbors through the communication network. Here, the following assumption is required:

Assumption 1. The underlying graph $\mathbb{G}$ of communication network is connected.

According to the problem (1), we give an equivalent problem by introducing the edge-based constraints:

$$\begin{aligned} &\min_{x} f(x) = \sum_{i=1}^{N} f_i(x_i) \\ &s.t.\ \sqrt{l_{ij}}(x_i - x_j) = 0,\ \forall \{i, j\} \in \mathbb{E}, \end{aligned} \tag{2}$$

where $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^{nN}$ and $x_i \in \mathbb{R}^n$ is an estimated solution of (1). From the definitions of the matrices $\mathcal{W}$ and $\mathcal{E}$, the constraints $\sqrt{l_{ij}}(x_i - x_j) = 0$ can be written as the compact form