Systems & Control Letters 99 (2017) 57–63

Contents lists available at ScienceDirect

Systems & Control Letters

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

Distributed resource allocation on dynamic networks in quadratic time

Thinh T. Doan [a,*], Alex Olshevsky [b]

[a] Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL, USA
[b] Department of Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, IL, USA

ARTICLE INFO

Article history:
Received 26 July 2015
Received in revised form 24 March 2016
Accepted 22 November 2016

Keywords:
Distributed optimization
Distributed resource allocation

ABSTRACT

We consider the problem of allocating a fixed amount of resource among nodes in a network when each node suffers a cost which is a convex function of the amount of resource allocated to it. We propose a new deterministic and distributed protocol for this problem. Our main result is that the associated convergence time for the global objective scales quadratically in the number of nodes on any sequence of time-varying undirected graphs satisfying a long-term connectivity condition.

1. Introduction

We consider the problem of optimally allocating a fixed amount of resource among n agents. Each agent suffers a convex cost as a function of the amount of resource allocated to it and the goal is to distribute the resource among the agents to minimize the total cost incurred. Sometimes the problem is described in terms of utilities, with each agent having a concave utility function and the goal being to maximize the total utility.

Our goal is to develop distributed protocols for this problem, meaning that nodes are only allowed to interact with neighbors in some graph or some time-varying sequence of graphs. Our motivation comes from potential applications in sensor networks, which regularly face the problems of optimally allocating communication bandwidth and computing power [1]. Furthermore, resource allocation is a simplification of the important "economic dispatch" problem wherein geographically distributed producers of electricity must coordinate to meet a fixed demand [2–4].

The problem has an old history dating back to the classic work of Arrow and Hurwicz [5]. The first algorithm which could be implemented in a distributed way was the "center-free" protocol of [6]. In the protocol of [6], each node increases the amount of resource allocated to it proportionally to the difference in gradients between its neighbors and itself. It was shown in [6] that, under appropriate technical conditions, this protocol will drive the amount of resource allocated to each node to the optimal value. The term "center-free" was originally meant to refer to the absence of any central coordinating authority, though in this paper we will use it to mean any update wherein nodes update the amount of resource by looking at gradient differences with neighbors. The work of [6] has spawned a number of modern follow-ups, including [6–11] as well as the current paper.

The paper [7] considered the resource allocation problem in the context of optimal distribution of a database among the nodes of the network; some modifications of the algorithm of [6] which used not only gradient differences but also the second derivatives of the cost functions were proposed. More recently, [8] studied the case when the cost functions are strongly convex and noted that the problem of optimal weight selection for center-free methods can be cast as a semidefinite program. The work of [9] analyzed a natural class of center-free methods on time-varying networks and provided a convergence analysis. The recent paper [10,11] studied the convergence rates of distributed protocols which repeatedly choose a random pair of neighboring nodes and perform a center-free update on that pair. Finally, the work of [12] used accelerated gradient methods to design distributed protocols for a more general problem.

Our focus in this paper is on designing protocols with good convergence speed. Specifically, we are interested at how the gap to the optimal objective value scales in the worst-case with iteration k and the number of nodes n in the system.

The best previously known results were provided in the antecedent papers [9,11]. Both papers considered the class of costs which have Lipschitz-continuous derivatives. The paper [11] considers schemes which randomly pick pairs of neighbors to perform a center-free update; if the pairs are chosen uniformly at random the convergence time implied by the results of [11] is $O(Ln^4/k)$ in

[*] Corresponding author.
E-mail addresses: ttdoan2@illinois.edu (T.T. Doan), aolshev2@illinois.edu (A. Olshevsky).

http://dx.doi.org/10.1016/j.sysconle.2016.11.006
0167-6911/© 2016 Elsevier B.V. All rights reserved.

expectation[1] on fixed graphs; here L is the largest of the Lipschitz constants of the derivatives of the cost functions. However, we note here that it is possible to shave off a factor of n off this bound by adjusting the probabilities in a graph-dependent way. The paper [9] does not give an explicit convergence rate for the objective, but gives a worst-case $O(LBn^3/k)$ rate for the decay of the average of *squared* gradient differences in the graph; here B is a constant which measures how long it takes for the time-varying graph sequence to reach connectivity. Improved rates were obtained in [13] and in [14] for a more general problem, but under the assumption that the graph is a fixed complete graph.

In this paper, we show a convergence rate of $O\left(LBn^2/k\right)$ for the objective under the same assumptions of Lipschitz-continuous derivatives in the more general setting of time-varying graphs. Additionally, when the costs are strongly convex, we demonstrate a geometric rate of $O\left(\left(1-\mu/(4Ln^2)\right)^{k/B}\right)$ where μ is the parameter of strong convexity. For both of these rates, the number of iterations until the objective is within ϵ of its optimal value scales quadratically with the number of nodes n. This is an improvement over the results described above, though we note that our protocol involves every node contacting its neighbors and performing an update at every step (which involves $O(|E(t)|)$ messages exchanged, where $E(t)$ is the set of edges at time t, and $O(n)$ updates); whereas [11] relied on only a pair of randomly chosen nodes updating at each step.

The remainder of this paper is organized as follows. We give a formal statement of the problem in Section 2. Our protocol is described in Section 3. The convergence analysis of the protocol is in Section 4. Finally, Section 5 describes the results of some simulations and we conclude in Section 6.

2. Problem formulation

In this paper, we study distributed protocols for the following minimization problem,

$$\min \sum_{i=1}^{n} f_i(x_i) \qquad (1)$$
$$\text{s.t.} \sum_{i=1}^{n} x_i = K.$$

We assume that there are n agents or nodes which we will index as $1, \ldots, n$, that $f_i : \mathbb{R} \to \mathbb{R}$ is a convex function known only to node i and $x_i \in \mathbb{R}$ is a variable stored by node i, and finally that K is some nonnegative number.

As remarked, this models a resource allocation problem among n agents: given a finite amount K of a certain resource, we must allocate it among agents $1, \ldots, n$ in an optimal way.

For simplicity, we introduce notation for the total objective function $F(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$, and the feasible set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i = K\}$.

We assume that a sequence of time-varying undirected graphs models the communication between the nodes. Specifically, we assume we are given a sequence of undirected graphs $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ with $\mathcal{V} = \{1, \ldots, n\}$; nodes i and j can send exchange messages at time k if and only if $(i,j) \in \mathcal{E}(k)$. We denote by $\mathcal{N}_i(k)$ the set of neighbors of node i at time k.

We make the following fairly standard assumption which ensures that the graph sequence $\mathcal{G}(k)$ satisfies a long-term connectivity property.

Assumption 1. There exists an integer $B \geq 1$ such that the undirected graph

$$(\mathcal{V}, \mathcal{E}(\ell B) \cup \mathcal{E}(\ell B + 1) \cup \cdots \cup \mathcal{E}((\ell+1)B - 1)) \qquad (2)$$

is connected for all nonnegative integers ℓ.

We will also be assuming that each local objective function $f_i(\cdot)$ is differentiable with Lipschitz continuous derivative.

Assumption 2. For each $i = 1, \ldots, n$, the function $f_i(\cdot)$ is differentiable everywhere and there exists a constant L_i such that

$$|f_i'(y_i) - f_i'(x_i)| \leq L_i |y_i - x_i|, \ \forall x_i, y_i \in \mathbb{R}.$$

Moreover, we will be assuming that there exists at least one optimal solution.

Assumption 3. There exists a vector $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ with $\mathbf{x}^* \in \mathcal{S}$ which achieves the minimum in problem (1).

We will use $\mathcal{X}^*$ to denote the set of optimal solutions to problem (1); the previous assumption ensures that $\mathcal{X}^*$ is not empty.

Finally, we will be assuming that our algorithm starts from a feasible point.

Assumption 4. $\mathbf{x}(0) \in \mathcal{S}$.

For the remainder of this paper, we will be assuming that Assumptions 1–4 hold without mention.

We conclude this section with a characterization of the points in the optimal set $\mathcal{X}^*$; the proof is immediate.

Proposition 1. *We have that $\mathbf{x} \in \mathcal{X}^*$ if and only if $\mathbf{x} \in \mathcal{S}$ and $f_i'(x_i) = f_j'(x_j)$ for all $i, j \in \{1, \ldots, n\}$.*

3. Main algorithm

In this section, we will introduce a distributed protocol, which we call the gradient balancing protocol, to solve problem (1). Before giving a statement of the algorithm, we provide some brief motivation for its form.

Previous protocols for problem (1) tended to be "center-free" updates [6,8,9,11] where node i updated as

$$x_i(k+1) = x_i(k) - \sum_{i \in \mathcal{N}_i(k)} w_{ij} \left(f_i'(x_i(k)) - f_j'(x_j(k))\right), \qquad (3)$$

where w_{ij} is a collection of nonnegative weights. The protocol of [11] had a different form but proceeded in the same spirit; in that protocol, edges were repeatedly chosen according to some probability distribution and a form of the above update was performed by the incident nodes.

The protocol we propose in this paper speeds up this update by employing some local "pruning" wherein each node tries to perform a version of Eq. (3), but only with the two nodes whose derivative is largest and smallest in its neighborhood. Thus nodes essentially ignore neighbors whose derivatives are close to their own. Intuitively, by focusing on nodes whose derivatives are far apart we increase the speed at which information propagates through the network. The idea has been previously used in [15] and is inspired by an algorithm from Chapter 7.4 of [16].

We now describe the steps node i executes at step k to update its value from $x_i(k)$ to $x_i(k+1)$. We assume that all nodes execute these steps synchronously, and furthermore that all four steps of the protocol given below can be executed before the graph changes from $\mathcal{G}(k)$ to $\mathcal{G}(k+1)$. Speaking informally, the protocol consists of each node repeatedly trying to "match" itself to the node in its neighborhood whose derivative is smallest and smaller than its own in order to perform a center-free update.

[1] The convergence rate in [11] is given in terms of the eigenvalues of a certain matrix; the quartic bound above follows by putting [11] together with the well-known fact that the smallest eigenvalue of the Laplacian of a connected, undirected graph on n is $\Omega(1/n^2)$.