Systems & Control Letters 107 (2017) 49–57

Contents lists available at ScienceDirect

Systems & Control Letters

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

Distributed optimal in-network resource allocation algorithm design via a control theoretic approach

Solmaz S. Kia

Department of Mechanical and Aerospace Engineering, University of California, Irvine 4200 Engineering Gateway, Irvine, CA 92697, United States

ARTICLE INFO

Article history:
Received 30 July 2016
Received in revised form 21 June 2017
Accepted 14 July 2017

Keywords:
Cooperative control
Distributed convex optimization
Distributed optimal resource allocation
Strongly connected and weight-balanced digraphs

ABSTRACT

In this paper, we consider an in-network optimal resource allocation problem with multiple demand equations. We propose a novel distributed continuous-time algorithm that solves the problem over strongly connected and weight-balanced digraph network topologies when the local cost functions are strongly convex. We also discuss the extension of our convergence guarantees to dynamically changing topologies. Finally, we show that if the network is an undirected connected graph, we can guarantee stability and convergence of our algorithm for problems involving local convex functions. This convergence guarantee is to a point in the set of minimizers of our optimal resource allocation problem. The design and analysis of our algorithm are carried out using a control theoretic approach. We demonstrate our results through a numerical example.

1. Introduction

This paper considers the problem of designing a distributed algorithm for an optimal resource allocation problem subject to a set of affine equality constraints over a network of N agents with communication and processing capabilities. In particular, each agent $i \in \{1, \ldots, N\}$ has a convex and differentiable local cost function $f^i : \mathbb{R} \to \mathbb{R}$. These agents are meeting some demands $\mathsf{b}_j \in \mathbb{R}, j \in \{1, \ldots, p\}$, through weighted contributions, in a way that the total cost $f(\mathbf{x}) = \Sigma_{i=1}^{N} f^i(x^i)$ is at its minimum. In other words, each agent $i \in \{1, \ldots, N\}$ seeks $\mathsf{x}_i^\star$, the ith element of $\mathbf{x}^\star$ given by

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \sum_{i=1}^{N} f^i(x^i), \quad \text{subject to} \tag{1a}$$

$$\omega_j^1 x^1 + \cdots + \omega_j^N x^N - \mathsf{b}_j = 0, \quad j \in \{1, \ldots, p\}, \tag{1b}$$

where, $\omega_j^i \in \mathbb{R}, i \in \{1, \ldots, N\}$, is the weight on the contribution of agent i to demand equation $j \in \{1, \ldots, p\}$. The weights $\{\omega_j^i\}_{j=1}^{p}$ of each agent $i \in \{1, \ldots, N\}$ are known to that agent. The aforementioned problem appears in many optimal decision making tasks such as economic dispatch over power networks [1,2], optimal routing [3] and economic systems [4].

Literature review: Our paper is related to a large recent literature on distributed algorithm design for solving a multi-agent optimization problem where the global cost function is a sum of local convex functions, each representing a private local cost only available to a single agent, subject to some convex constraints. Some of the recent literature on distributed optimization algorithm design includes distributed algorithms implemented both in discrete-time [5–9] and continuous-time [10–14]. Although some of these algorithms can solve the optimal resource allocation problem (1), they require each agent to keep and evolve a copy of the global decision variable of the problem which is of order N, where N is the size of network. Such a requirement is costly and unnecessary for problem (1), as the agents only need to obtain their own respective component of the global decision variable. Distributed optimization algorithms that specifically target the optimal resource allocation problem (1) are presented in [15] in discrete-time form, and [2,16] in continuous-time form. These algorithms require the agents to keep and evolve only their respective component of the global decision variable. However, these algorithms all can solve the optimal allocation problem (1) subject to single unweighted demand equation, i.e., $\omega_1^i = 1$, $i \in \{1, \ldots, N\}$ and $p = 1$ in (1b). Also, these algorithms require the agents to transmit the gradient of their local cost functions to their neighbors, which makes these algorithms less favorable for privacy-sensitive applications. The composition of our algorithm is inspired by the multi-time scale singularly perturbed systems in control theory (cf. [17]). Singularly perturbed distributed algorithms are used in [12] for unconstrained in-network convex optimization, and in [18] for dynamic consensus problem over networked systems.

Statement of contributions: We propose a novel continuous-time distributed algorithm to solve the optimal resource allocation problem (1) over networked systems. We show that our algorithm

E-mail address: solmaz@uci.edu.

http://dx.doi.org/10.1016/j.sysconle.2017.07.012
0167-6911/

converges over strongly connected and weight-balanced digraphs if the local cost functions are strongly convex. Such guarantees also hold for time-varying strongly connected and weight-balanced digraphs with piecewise constant adjacency matrices if the gradients of all local cost functions are globally Lipschitz. When the communication graph is an undirected connected graph, the convergence is guaranteed for convex local cost functions, as well. Our convergence guarantee is to a point in the optimizer set. The composition of our algorithm is inspired by the singular perturbation systems in control theory. The idea behind this composition is that an average consensus algorithm creates a local copy of the left hand side of the equality constraint (1b) at each agent. This way, every agent can create a local copy of its respective part in a centralized saddle-point dynamical solver used in the literature to solve the optimization problem (1). In the resulted algorithm, each agent is only required to keep a copy of its own local decision variable. Also, agents are not required to share the gradient of their local cost functions with their neighbors. We use Lyapunov and invariant set analysis to study the convergence and stability of our proposed algorithm. A preliminary work related to our work has appeared in [19].

2. Preliminaries

This section presents our notations, definitions, a review of relevant algebraic graph theory, and the average consensus algorithm of [20].

2.1. Notations

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$, respectively, be the set of real, non-negative real, and positive real numbers. We let $\mathbf{1}_n$ (resp. $\mathbf{0}_n$) denote the vector of n ones (resp. n zeros), and denote by $\mathbf{I}_n$ the $n \times n$ identity matrix. When clear from the context, we do not specify the matrix dimensions. We denote the standard Euclidean norm of vector $\mathbf{x} \in \mathbb{R}^n$ by $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$. We denote the induced 1-norm, ∞-norm and spectral norm of a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ by, respectively, $\|\mathbf{A}\|_1$, $\|\mathbf{A}\|_\infty$ and $\|\mathbf{A}\|$. In a network of N agents, to distinguish and emphasize that a variable is local to an agent $i \in \{1, \dots, N\}$, we use superscripts, e.g., $f^i(x^i)$ is the local function of agent i evaluated at its own local state x^i. Moreover, if $\mathbf{p}^i \in \mathbb{R}^d$ is a variable of agent $i \in \{1, \dots, N\}$, the aggregated $\mathbf{p}^i$'s of the network is the vector $\mathbf{p} = [\,{\mathbf{p}^1}^\top, \cdots, {\mathbf{p}^N}^\top]^\top \in (\mathbb{R}^d)^N$.

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex (resp. strictly convex) over a convex set $C \subseteq \mathbb{R}^d$ iff $(\mathbf{z}-\mathbf{x})^\top(\nabla f(\mathbf{z}) - \nabla f(\mathbf{x})) \geq 0$ (resp. $(\mathbf{z}-\mathbf{x})^\top(\nabla f(\mathbf{z}) - \nabla f(\mathbf{x})) > 0$ whenever $\mathbf{x} \neq \mathbf{z}$) for all $\mathbf{x}, \mathbf{z} \in C$, and it is *m-strongly convex* ($m \in \mathbb{R}_{>0}$) iff $(\mathbf{z}-\mathbf{x})^\top(\nabla f(\mathbf{z}) - \nabla f(\mathbf{x})) \geq m\|\mathbf{z}-\mathbf{x}\|^2$, for all $\mathbf{x}, \mathbf{z} \in C$. A function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz with constant $M \in \mathbb{R}_{>0}$, or simply *M-Lipschitz*, over a set $C \subseteq \mathbb{R}^d$ iff $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq M\,\|\mathbf{x}-\mathbf{y}\|$, for $\mathbf{x}, \mathbf{y} \in C$. Function f is *globally Lipschitz* if it is M-Lipschitz over $\mathbb{R}^d$. Moreover, it is *locally Lipschitz* on $\mathbb{R}^d$ if for every point $\mathbf{x} \in \mathbb{R}^d$ there exists a $M_x \in \mathbb{R}_{>0}$ such that $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq M_x\,\|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{y}$ in an open and connected neighborhood of $\mathbf{x}$.

2.2. Graph theory

We briefly review basic concepts from algebraic graph theory following [21]. A *digraph*, is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the *node set* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*. An edge from i to j, denoted by (i, j), means that agent j can send information to agent i. For an edge $(i, j) \in \mathcal{E}$, i is called an *in-neighbor* of j and j is called an *out-neighbor* of i. A graph is *undirected* if $(i, j) \in \mathcal{E}$ anytime $(j, i) \in \mathcal{E}$. A *directed path* is a sequence of nodes connected by edges. A digraph is *strongly connected* if for every pair of nodes there is a directed path connecting them. A *weighted digraph* is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $(\mathcal{V}, \mathcal{E})$ is a digraph and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a weighted *adjacency* matrix such that $\mathsf{a}_{ij} > 0$ if $(i, j) \in \mathcal{E}$ and $\mathsf{a}_{ij} = 0$, otherwise. A weighted digraph is *undirected* if $\mathsf{a}_{ij} = \mathsf{a}_{ji}$ for all $i, j \in \mathcal{V}$. A *connected graph* is a strongly connected and undirected graph. The *weighted in-* and *out-degrees* of a node i are, respectively, $\mathsf{d}^i_{\text{in}} = \Sigma_{j=1}^N \mathsf{a}_{ji}$ and $\mathsf{d}^i_{\text{out}} = \Sigma_{j=1}^N \mathsf{a}_{ij}$. A digraph is *weight-balanced* if at each node $i \in \mathcal{V}$ the weighted out-degree and weighted in-degree coincide. Any connected graph is weight-balanced. The *(out-) Laplacian* matrix is $\mathbf{L} = \mathbf{D}^{\text{out}} - \mathbf{A}$, where $\mathbf{D}^{\text{out}} = \text{Diag}(\mathsf{d}^1_{\text{out}}, \cdots, \mathsf{d}^N_{\text{out}}) \in \mathbb{R}^{N\times N}$. Note that $\mathbf{L}\mathbf{1}_N = \mathbf{0}$. A digraph is weight-balanced iff $\mathbf{1}_N^T \mathbf{L} = \mathbf{0}$. We let $\{\lambda_i\}_{i=1}^N$ and $\{\hat{\lambda}_i\}_{i=1}^N$, respectively, be the set of eigenvalues of $\mathbf{L}$ and $\text{Sym}(\mathbf{L}) = (\mathbf{L} + \mathbf{L}^\top)/2$. Based on the structure of $\mathbf{L}$, at least one of the eigenvalues of $\mathbf{L}$ is zero ($\lambda_1 = 0$) and the rest of them have nonnegative real parts. For a strongly connected and weight-balanced digraph, zero is a simple eigenvalue of both $\mathbf{L}$ and $\text{Sym}(\mathbf{L})$. Moreover, we have

$$0 < \hat{\lambda}_2 \mathbf{I} \leq \mathbf{R}^\top \,\text{Sym}(\mathbf{L})\mathbf{R} \leq \hat{\lambda}_N \mathbf{I}, \tag{2}$$

where $\hat{\lambda}_2$ and, $\hat{\lambda}_N$ are, respectively, the smallest non-zero eigenvalue and maximum eigenvalue of $\text{Sym}(\mathbf{L})$. Here, $\mathbf{R} \in \mathbb{R}^{N\times(N-1)}$ along with $\mathbf{r} \in \mathbb{R}^N$ satisfies

$$\mathbf{r} = \frac{1}{\sqrt{N}}\mathbf{1}_N,\;\; \mathbf{r}^\top\mathbf{R} = \mathbf{0},\;\; \mathbf{R}^\top\mathbf{R} = \mathbf{I}_{N-1},\;\; \mathbf{R}\mathbf{R}^\top = \mathbf{I}_N - \mathbf{r}\mathbf{r}^\top. \tag{3}$$

For connected graphs $\hat{\lambda}_i = \lambda_i$, $i \in \mathcal{V}$, therefore, $0 < \lambda_2 \mathbf{I} \leq \mathbf{R}^\top\mathbf{L}\mathbf{R} \leq \lambda_N \mathbf{I}$.

2.3. Average consensus algorithm

Let $\mathcal{G}$ be a strongly connected and weight-balanced digraph of N agents. Assume each node $i \in \mathcal{V}$ has access to a static reference input $\mathbf{r}^i \in \mathbb{R}^p$. [20] shows that for $\beta \in \mathbb{R}_{>0}$, if each agent $i \in \mathcal{V}$, implements

$$\dot{\mathbf{v}}^i = \beta \sum_{j=1}^N \mathsf{a}_{ij}(\mathbf{y}^i - \mathbf{y}^j), \tag{4}$$

$$\dot{\mathbf{y}}^i = -(\mathbf{y}^i - \mathbf{r}^i) - \beta \sum_{j=1}^N \mathsf{a}_{ij}(\mathbf{y}^i - \mathbf{y}^j) - \mathbf{v}^i,$$

starting at $\mathbf{y}^i(0), \mathbf{v}^i(0) \in \mathbb{R}^p$, $\sum_{j=1}^N \mathbf{v}^j(0) = \mathbf{0}$, then as $t \to \infty$, its state $\mathbf{y}^i$ converges to $\frac{1}{N}\sum_{j=1}^N \mathbf{r}^j$ exponentially fast.

3. Problem statement

We consider the optimal resource allocation problem (1) over a network of N agents interacting over a digraph $\mathcal{G}$, and under the following assumption.

Assumption 1. Matrix $\Omega = [\boldsymbol{\omega}^1, \cdots, \boldsymbol{\omega}^N]$, where $\boldsymbol{\omega}^i = [\omega^i_1, \cdots, \omega^i_p]^\top$, $i \in \mathcal{V}$, is full row rank. Moreover, the optimization problem (1) has a finite optimum $f^\star = f(\mathbf{x}^\star)$. Finally, ∇f^i, $i \in \mathcal{V}$, is locally Lipschitz on $\mathbb{R}$.

The first part of Assumption 1 ensures that the feasible set of optimization problem (1) is non-empty and the problem has a finite minimizer in the feasible set. Local Lipschitzness of ∇f^i, $i \in \mathcal{V}$, guarantees existence and uniqueness of the solutions of the dynamical solvers that we study in this paper for problem (1) (cf. [22, Theorem 3.3])—these solvers use ∇f^i, $i \in \mathcal{V}$.