Systems & Control Letters 91 (2016) 36–42

Contents lists available at ScienceDirect

Systems & Control Letters

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

Distributed partitioning algorithms for multi-agent networks with quadratic proximity metrics and sensing constraints

E. Bakolas

Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712-1221, USA

ARTICLE INFO

Article history:
Received 29 May 2015
Received in revised form
20 September 2015
Accepted 5 February 2016
Available online 15 March 2016

Keywords:
Voronoi diagrams
Distributed partitioning algorithms
Sensing constraints

ABSTRACT

We consider a class of Voronoi-like partitioning problems, in which a multi-agent network seeks to subdivide a subset of an affine space into a finite number of cells in the presence of sensing constraints. The cell of this subdivision that is assigned to a particular agent consists exclusively of points that can be sensed by this agent and are closer to it than to any other agent that can also sense them. The proximity between an agent and an arbitrary point is measured in terms of a non-homogeneous quadratic (generalized) distance function, which does not, in general, enjoy the triangle inequality and the symmetry property. One of the consequences of this fact is that the structure of the sublevel sets of the utilized proximity metric does not conform with that of the sensing region of an agent. Due to this mismatch, it is possible that a point may be assigned to an agent which is different from its "nearest" agent simply because the nearest agent cannot sense this point, unless special care is taken. We propose a distributed partitioning algorithm that enables each agent to compute its own cell independently from the other agents when the only information available to it is the positions and the velocities of the agents that lie inside its sensing region. The algorithm is based on an iterative process that adjusts the size of the sensing region of each agent until the associated cell of the latter corresponds to the intersection of its sensing region with the cell that would have been assigned to it in the absence of sensing constraints. The correctness of the proposed distributed algorithm, which successfully handles the aforementioned issues, is studied in detail.

1. Introduction

In this work, we consider a spatial partitioning problem for a multi-agent network in the presence of sensing constraints. The region to be partitioned is a subset of an affine subspace which is comprised of points that can be reached by the agents with zero terminal velocity (*terminal manifold* of the multi-agent network). It is assumed that each agent can measure its distance from an arbitrary point in the terminal manifold by means of a non-homogeneous quadratic (generalized) distance function, provided this point lies in its sensing region. In addition, it is assumed that each agent can only sense the velocities and the positions of its teammates that lie within its sensing region. The solution to this partitioning problem corresponds to a collection of non-overlapping cells that are assigned to different agents. Specifically, the cell assigned to a particular agent will consist exclusively of points that are (1) within the agent's sensing region and (2) closer, in terms of the utilized proximity metric, to this agent than to any other agent from the same network that can also sense them.

Literature Review: Partitioning problems are becoming very relevant to several classes of sensing and control problems involving networks of autonomous agents and mobile sensors [1–10]. Specifically, partitioning algorithms can provide such networks, which are typically assigned with multiple and spatially distributed tasks, with the necessary means to "optimize the quality of service" they provide, according to the authors of [1,11].

In our previous work, we have proposed a new class of partitioning problems in which the proximity metric corresponds to the optimal value function of a quadratic optimal control problem [12,13]. This class of spatial partitions corresponds to a special class of generalized Voronoi diagrams (see [14,15] and references therein), given that the utilized proximity metric in [12,13] is different than those used in standard Voronoi diagram problems. To address this class of problems, we have proposed algorithms which are decentralized in the sense that they enable each agent to compute its own cell independently from its teammates without utilizing, for instance, a common spatial grid. The main caveat of the approach proposed in [12,13] is that its

E-mail address: bakolas@austin.utexas.edu.

http://dx.doi.org/10.1016/j.sysconle.2016.02.005
0167-6911/

decentralized implementation hinges upon the assumption that each agent knows the positions and the velocities of all the other agents. Actually, it suffices to assume that each agent knows the positions and the velocities of its neighbors in the topology of the Voronoi-like partition. However, in the latter case, the required information about the neighboring relations among the agents cannot be available to an agent unless the latter knows the whole solution to the partitioning problem a priori. In the presence of sensing constraints, such an assumption is practically impossible to be verified.

Main Contributions and Challenges: The main contribution of this work is the presentation of a distributed algorithm for a class of partitioning problems involving multi-agent networks, which, in contrast with some of our previous results in this class of problems [12,13], accounts explicitly for the presence of sensing constraints. In the proposed framework, each agent is assumed to know only the positions and the velocities of the agents that lie in its sensing region. Following the approach that has been proposed in the literature for the distributed computation of standard Voronoi diagrams [16,1] (in which the proximity metric is the Euclidean distance), we will relax the sensing constraints by allowing each agent to adjust the size of its sensing region. The objective here is twofold. First, the sensing region of an agent should be large enough to allow it to infer the positions and velocities of those of its teammates in the network that are necessary for the independent computation of its own cell. Second, the computed cell should be a *consistent truncation* of the cell that would be assigned to it in the absence of sensing constraints. Here, the term "consistent truncation" describes the situation in which the intersection of the sensing region of an agent with its cell in the absence of sensing constraints coincides with its assigned cell in the presence of sensing constraints.

In the problem we consider herein, the distributed computation of the Voronoi-like partition poses new challenges, which cannot be tackled by means of the available techniques used for the distributed computation of standard Voronoi partitions [16,1,17,7]. This is mainly because the proximity metric utilized here, which is a non-homogeneous quadratic function, does not enjoy "nice properties" such as the triangle inequality and the symmetry property in contradistinction with the Euclidean distance. A consequence of this fact is that the structure of the sub-level sets of the proximity metric of an agent, which are ellipsoids centered at a point that is different, in general, from the agent's location, does not match that of its sensing region, which is a ball centered at the agent's location. Because of this mismatch, it is possible that points may be assigned to an agent that is different from their nearest one, in terms of the utilized proximity metric, because the latter agent cannot sense them, unless special care is taken. The proposed algorithm, whose correctness is analyzed in detail, addresses successfully all the aforementioned issues.

Organization of the paper: The rest of the paper is organized as follows. Section 2 presents the formulation of the partitioning problem in the presence of sensing constraints. A distributed partitioning algorithm along with a detailed analysis of its correctness are presented in Section 3. Section 4 presents numerical simulations, and finally, Section 5 concludes the paper with a summary of remarks.

2. Formulation and analysis of the partitioning problem in the presence of sensing constraints

2.1. Notation

We denote by $\mathbb{R}^m$ the set of m-dimensional real vectors. We denote by $\mathbb{R}_{\geq 0}$ and $\mathbb{Z}_{\geq 0}$, respectively, the sets of non-negative real numbers and integers. We write $|\boldsymbol{\alpha}|$ to denote the 2-norm of a vector $\boldsymbol{\alpha} \in \mathbb{R}^m$. We write $\mathbf{P} = \mathbf{P}^{\mathrm{T}} \succ \mathbf{0}$ to denote the fact that a square (symmetric) matrix is positive definite. Furthermore, we denote by $\lambda_{\min}(\mathbf{P})$ and $\lambda_{\max}(\mathbf{P})$ the minimum and the maximum eigenvalues of a symmetric matrix $\mathbf{P}$, respectively. Similarly, the minimum and the maximum singular values of a matrix $\mathbf{A}$ are denoted by $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$, respectively. In addition, $\mathrm{bd}(S)$ and $\mathrm{int}(S)$ denote, respectively, the boundary and the interior of a set $S \subset \mathbb{R}^m$. The relative interior of a set S will be denoted by $\mathrm{rint}(S)$. The closed ball of radius ϱ around a point $\boldsymbol{x} \in \mathbb{R}^m$ will be denoted by $\overline{\mathcal{B}}(\boldsymbol{x}; \varrho)$. Finally, $\mathcal{L}^2([0, \tau], \mathbb{R}^m)$ denotes the space of square integrable functions $g : [0, \tau] \mapsto \mathbb{R}^m$, for a given $\tau > 0$.

2.2. Problem setup

We are given a network of n agents which are initially located at n distinct points, $\bar{\boldsymbol{x}}_i \in \mathbb{R}^2$, with prescribed initial velocities, $\bar{\boldsymbol{v}}_i \in \mathbb{R}^2$, where $i \in \mathcal{I}_n := \{1, \ldots, n\}$. We denote by $\overline{\mathcal{X}} := \{\bar{\boldsymbol{x}}_i \in \mathbb{R}^2,\ i \in \mathcal{I}_n\}$ and $\overline{\mathcal{V}} := \{\bar{\boldsymbol{v}}_i \in \mathbb{R}^2,\ i \in \mathcal{I}_n\}$, respectively, the sets of initial positions and initial velocities of all the agents. The motion of the ith agent from the network, where $i \in \mathcal{I}_n$, is described by the following set of equations:

$$\dot{\boldsymbol{z}}_i = \mathbf{A}(t)\boldsymbol{z}_i + \mathbf{B}(t)\boldsymbol{u}_i(t), \qquad \boldsymbol{z}_i(0) = \bar{\boldsymbol{z}}_i, \tag{1}$$

where $\boldsymbol{z}_i := [\boldsymbol{x}_i^{\mathrm{T}},\ \boldsymbol{v}_i^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^4$ and $\bar{\boldsymbol{z}}_i := [\bar{\boldsymbol{x}}_i^{\mathrm{T}},\ \bar{\boldsymbol{v}}_i^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^4$ denote, respectively the state of the ith vehicle (concatenation of position and velocity vectors) at time t and $t = 0$; the set of initial states of all the agents is denoted by $\overline{\mathcal{Z}} := \{\bar{\boldsymbol{z}}_i \in \mathbb{R}^4,\ i \in \mathcal{I}_n\}$. Moreover, $\boldsymbol{u}_i(\cdot) \in \mathcal{L}^2([0, \tau], \mathbb{R}^2)$ denotes the control input of the ith agent. In addition, $\mathbf{A}(\cdot)$ and $\mathbf{B}(\cdot)$ are continuous matrix-valued functions of time and can be defined, for instance, as in [13], in which case, $\mathbf{A}(t) := \begin{bmatrix} \mathbf{0}_2 & \mathbf{I}_2 \\ -\mathbf{K}(t) & -\mathbf{C}(t) \end{bmatrix}$, $\mathbf{B}(t) := \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{H}(t) \end{bmatrix}$, where $\mathbf{I}_2$ and $\mathbf{0}_2$ are the identity and the zero 2×2 matrices, respectively, and $\mathbf{K}(\cdot)$, $\mathbf{C}(\cdot)$ and $\mathbf{H}(\cdot)$ are continuous matrix-valued functions of time; in addition, $\mathbf{H}(t)$ is a non-singular 2×2 matrix for all $t \geq 0$. Finally, the *terminal manifold*, which is denoted by $\mathcal{X}_0$, is taken to be a two-dimensional affine subspace embedded in $\mathbb{R}^4$, which consists of all the positions that can be reached with a zero terminal velocity, that is, $\mathcal{X}_0 := \{\boldsymbol{z} = [\boldsymbol{x}^{\mathrm{T}},\ \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^4 : \boldsymbol{v} = 0\}$.

Following [13], we will be measuring the distance between the ith agent and an arbitrary point $\boldsymbol{z}(\boldsymbol{x}) := [\boldsymbol{x}^{\mathrm{T}},\ 0]^{\mathrm{T}}$ in the terminal manifold $\mathcal{X}_0$ by means of the minimum control effort required for the former to reach the latter. In particular, let $\tau > 0$ and let $\mathcal{U}(\boldsymbol{x}; \tau, \bar{\boldsymbol{z}}_i) := \{\boldsymbol{u}_i(\cdot) \in \mathcal{L}^2([0, \tau], \mathbb{R}^2) : \boldsymbol{z}_i(\tau; \bar{\boldsymbol{z}}_i, \boldsymbol{u}_i(\cdot)) = [\boldsymbol{x}^{\mathrm{T}},\ 0]^{\mathrm{T}}\}$ where $\boldsymbol{z}_i(\cdot; \bar{\boldsymbol{z}}_i,\ \boldsymbol{u}_i(\cdot))$ denotes the solution to the initial value problem given in (1) for a given input $\boldsymbol{u}_i(\cdot)$. It can be shown that if $\mathcal{U}(\boldsymbol{x}; \tau, \bar{\boldsymbol{z}}_i) \neq \varnothing$, which is always true when the system (1) is controllable at $t = \tau$, then the minimum control effort required to steer the system (1) from $\bar{\boldsymbol{z}}_i$ to $\boldsymbol{z}(\boldsymbol{x}) := [\boldsymbol{x}^{\mathrm{T}},\ 0]^{\mathrm{T}}$ at time $t = \tau$, which is denoted by $J^{\circ}(\boldsymbol{x}; \tau, \bar{\boldsymbol{z}}_i)$, where

$$J^{\circ}(\boldsymbol{x}; \tau, \bar{\boldsymbol{z}}_i) := \min_{\boldsymbol{u}_i(\cdot) \in \mathcal{U}(\boldsymbol{x}; \tau, \bar{\boldsymbol{z}}_i)} \int_0^{\tau} \tfrac{1}{2} |\boldsymbol{u}_i(t)|^2 \mathrm{d}t,$$

satisfies the following equation:

$$J^{\circ}(\boldsymbol{x}; \tau, \bar{\boldsymbol{z}}_i) = \langle \boldsymbol{x} - \boldsymbol{q}(\tau, \bar{\boldsymbol{z}}_i), \mathbf{P}(\tau)(\boldsymbol{x} - \boldsymbol{q}(\tau, \bar{\boldsymbol{z}}_i)) \rangle + \delta(\tau, \bar{\boldsymbol{z}}_i), \tag{2}$$

where $\mathbf{P}(\tau)$ is a positive definite 2×2 matrix, that is, $\mathbf{P}(\tau) = \mathbf{P}^{\mathrm{T}}(\tau) \succ 0$, $\boldsymbol{q}(\tau, \bar{\boldsymbol{z}}_i)$ is a two-dimensional column vector and $\delta(\tau, \bar{\boldsymbol{z}}_i)$ is a non-negative number. Note that the matrix $\mathbf{P}(\tau)$ does not depend on any parameter besides the final time τ and is solely determined by the solution to the optimal control problem. Similarly, $\boldsymbol{q}(\tau, \bar{\boldsymbol{z}}_i)$ and $\delta(\tau, \bar{\boldsymbol{z}}_i)$ depend only on the final time τ and the initial state of the ith agent. In other words, no parameter selection, which would potentially put in question the distributed character of the algorithmic tools that will be introduced later on, is required. Moreover, in order to better illustrate the connection of