Automatica 52 (2015) 227–231

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Continuous graph partitioning for camera network surveillance☆

Domenica Borra[a], Fabio Pasqualetti[b], Francesco Bullo[c]

[a] Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy

[b] Department of Mechanical Engineering, University of California at Riverside, Riverside, USA

[c] Center for Control, Dynamical Systems, and Computation, University of California at Santa Barbara, Santa Barbara, USA

ARTICLE INFO

Article history:
Received 6 April 2014
Received in revised form
21 July 2014
Accepted 3 November 2014
Available online 26 December 2014

Keywords:
Graph partitioning
Constrained optimization
Camera network
Distributed control

ABSTRACT

In this note we discuss a novel graph partitioning problem, namely continuous graph partitioning, and we discuss its application to the design of surveillance trajectories for camera networks. In continuous graph partitioning, each edge is partitioned in a continuous fashion between its endpoint vertices, and the objective is to minimize the largest load among the vertices. We show that the continuous graph partitioning problem is convex and non-differentiable, and we characterize a solution amenable to distributed computation. The continuous graph partitioning problem naturally arises in the context of camera networks, where intruders appear at arbitrary locations and times, and the objective is to design camera trajectories for quickest detection of intruders. Finally, we propose a surveillance strategy for networks of PTZ cameras and we characterize its performance.

1. Introduction

Autonomous camera networks are becoming the leading technology for the surveillance of human activities in civil and military applications (Bhanu, Ravishankar, Roy-Chowdhury, Aghajan, & Terzopoulos, 2010). Besides computer vision and pattern recognition difficulties, the design of efficient algorithms for the cameras to autonomously and distributively complete tracking, surveillance, and recognition tasks remains one of the main challenges.

In this note we focus on the problem of detecting static intruders by means of a network of autonomous PTZ cameras. We assume the cameras to move their *field of view* (f.o.v.) to cooperatively surveil the whole environment, and we develop algorithms for the cameras to self-organize, coordinate and detect intruders in the shortest amount of time. To this aim, we present a novel graph partitioning problem, namely *continuous graph partitioning*, which is used to optimally assign regions of competence to each camera.

Related work. The recent literature on coordination problems in camera networks mobile robotics is of relevance to this work. In Baseggio, Cenedese, Merlo, Pozzi, and Schenato (2010), Carli, Cenedese, and Schenato (2011) and Spindler, Pasqualetti, and Bullo (2012) distributed algorithms are proposed for PTZ cameras to partition a one-dimensional environment, and to synchronize along a trajectory with minimum worst-case detection time of intruders. We improve the results along these directions by, for instance, developing cameras trajectories and partitioning methods for general environment topologies. In mobile robotics, the patrolling problem consists of scheduling the motion of a team of autonomous agents in order to detect intruders or important events, e.g., see Alberton, Carli, Cenedese, and Schenato (2012), Baseggio et al. (2010), Machado, Ramalho, Zucker, and Drogoul (2003) and Pasqualetti, Franchi, and Bullo (2012). The patrolling problem and the problem considered in this paper significantly differ. In fact, cameras are fixed at predetermined locations, and their f.o.v.s must lie within the cameras visibility constraints. Instead, robots are usually allowed to travel the whole environment, and are usually not subject to visibility constraints. Consequently, algorithms developed for teams of robots are, in general, not applicable in the present setup. Similarly, algorithms developed in the computer science community for graph-clearing and graph-search do not extend to our scenario (Kehagias, Hollinger, & Singh, 2009; Kolling & Carpin, 2010; Parsons, 1978).

In this work we present algorithms for graph partitioning. Our graph partitioning problem differs from classical setups, e.g., see

☆ This work was supported in part by NSF grants IIS-0904501, CNS-1035917, and ECCS 1405330, and by ARO grant W911NF-11-1-0092. A preliminary and shortened version of this work appeared in Borra, Pasqualetti, and Bullo (2012). The material in this paper was presented at the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12), September 14–15, 2012, Santa Barbara, California, USA. This paper was recommended for publication in revised form by Associate Editor Wei Ren under the direction of Editor Frank Allgöwer.

E-mail addresses: domenica.borra@polito.it (D. Borra), fabiopas@engr.ucr.edu (F. Pasqualetti), bullo@engineering.ucsb.edu (F. Bullo).

http://dx.doi.org/10.1016/j.automatica.2014.11.017
0005-1098/

Andreev and Racke (2006), Arkin, Hassin, and Levin (2006) and Even, Garg, Könemann, Ravi, and Sinha (2004); Even, Naor, Rao, and Schieber (1997). In fact, classic graph partitioning problems require the partitioning of the vertices or edges. Instead, we formulate a continuous graph partitioning problem, where the graph is a physical entity, and the partition is obtained by partitioning each edge among its endpoint vertices. Continuous graph partitioning problems arise in different application domains. For instance, if each edge of a graph represents a task to be accomplished by some processors, then our algorithms can be used for dynamic load balancing in multiprocessor networks (Cybenko, 1989; Lüling, Monien, & Ramme, 1991).

Paper contributions. The main contributions of this work are as follows. First, we propose the *continuous graph partitioning problem*, where a partition of a weighted graph is obtained by splitting the graph edges, and the cost of a partition equals the longest length of its parts (Section 2). We show that the continuous graph partitioning problem is convex and non-differentiable, and we characterize its solutions. Then, we derive an equivalent convex and differentiable partitioning problem, which is amenable to distributed implementation.

Second, we design trajectories for networks of autonomous PTZ cameras for the detection of static intruders (Section 4). We model the environment and the camera network by means of a robotic roadmap, and we formalize the worst-case detection time of intruders as performance criterium. We show that, for tree and ring roadmaps, cameras trajectories with minimum worst-case detection time can be designed by solving a continuous graph partitioning problem. For general cyclic roadmaps, our trajectories based on continuous partitions are proved to be optimal up to a factor 2.

Third and finally, we design a distributed algorithm for the computation of an optimal cameras trajectories based on continuous graph partitioning. Our algorithm relies on asymmetric broadcast communication, in which at each iteration only one camera updates its state by using local information from its neighboring cameras.

2. Continuous partitions of weighted graphs

Graph partitioning is a classic problem in computer science and robotics (Castellano, Fortunato, & Loreto, 2009). In this section we introduce a novel graph partitioning problem, namely *continuous graph partitioning*, in which each edge is partitioned in a continuous fashion between its endpoint vertices. As we discuss later, continuous graph partitioning finds application in camera networks and in robotics applications.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected weighted graph, where $\mathcal{V} = \{1, \dots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the vertex and edge sets, respectively. We associate a point $v_i \in \mathbb{R}^2$ with each vertex $i \in \mathcal{V}$, and we let $[v_i, v_j]$ denote the segment joining v_i and v_j. Let $\ell_{ij} = \|v_i - v_j\|_2$ be the weight associated with the edge $(i, j) \in \mathcal{E}$. Finally, define the neighbors of node i as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$.

A *continuous partition* of the weighted graph $\mathcal{G}$ is a set $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ where, for $i \in \{1, \dots, n\}$,

$$\mathcal{P}_i = \bigcup_{v_j \in \mathcal{N}_i} [v_i, \; v_{ij}], \tag{1}$$

and v_{ij} is a point along the segment $[v_i, v_j]$ defined by the parameter $\alpha_{ij} \in [0, 1]$ as

$$v_{ij} = \begin{cases} v_i + \alpha_{ij}(v_j - v_i), & \text{if } i < j, \\ v_i + (1 - \alpha_{ji})(v_j - v_i), & \text{if } i > j. \end{cases} \tag{2}$$

Let $\boldsymbol{\alpha} = [\alpha_{ij}]$ be the vector containing all parameters α_{ij}, and notice that the partition $\mathcal{P}$ is entirely specified by the vector $\boldsymbol{\alpha}$. Each undirected edge (i, j) is associated with a parameter α_{ij}. We adopt the convention

$$\alpha_{ji} = 1 - \alpha_{ij}, \quad \text{if } i < j.$$

For notational convenience, we sometimes identify a partition with its parameters vector.

The *length*, or *cost*, of the continuous partition $\mathcal{P}$ is denoted as

$$\mathcal{L}(\mathcal{P}) = \max\{L_1, \dots, L_n\}, \tag{3}$$

where L_i is the sum of the lengths of the segments in $\mathcal{P}_i$, that is,

$$L_i = \sum_{j>i} \alpha_{ij}\ell_{ij} + \sum_{j<i} (1 - \alpha_{ji})\ell_{ij}. \tag{4}$$

Let L be the vector of L_i.

Let $A \in \mathbb{R}^{n \times |\mathcal{E}|}$ be the weighted incidence matrix of $\mathcal{G}$, where, for each edge $e = (v_i, v_j) \in \mathcal{E}_c$,

$$A_{i,e} = \begin{cases} \ell_{ij}, & \text{if } i < j, \\ -\ell_{ij}, & \text{if } i > j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Define the *incidence vector* $\boldsymbol{b} \in \mathbb{R}^n$ as

$$\boldsymbol{b}_i = \sum_{i>j} \ell_{ij}, \tag{6}$$

and notice that $L = A\boldsymbol{\alpha} + \boldsymbol{b}$. Additionally, it can be verified that $\mathcal{L}(\mathcal{P}) = \|A\boldsymbol{\alpha} + b\|_\infty$ and, for every $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{E}|}$,

$$\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_1 = \sum_{(i,j)\in\mathcal{E}} \ell_{ij}.$$

Let $\mathbf{0}$ and $\mathbf{1}$ be the vectors of all zeros and ones, respectively. The *min–max continuous partitioning problem* is stated as follows.

Problem 1 (*Continuous Min–Max Partitioning*)**.** For a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, determine a continuous partition $\boldsymbol{\alpha}_\infty^*$ satisfying

$$\|A\boldsymbol{\alpha}_\infty^* + \boldsymbol{b}\|_\infty = \min_{\underline{\boldsymbol{\alpha}} \le \boldsymbol{\alpha} \le \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty, \tag{7}$$

where A and $\boldsymbol{b}$ are as in (5) and (6), for some constraints vectors $\mathbf{0} \le \underline{\boldsymbol{\alpha}} \le \overline{\boldsymbol{\alpha}} \le \mathbf{1}$.

It should be observed that (7) is a convex minimization problem, for which efficient centralized solvers exist (Boyd & Vandenberghe, 2004). On the other hand, since (7) is not differentiable, distributed solvers may be difficult to implement. We next derive an equivalent differentiable minimization problem, which is amenable to distributed implementation. In Problem 1 the vectors $\underline{\alpha}$ and $\overline{\alpha}$ represent possible constraints on the partition as dictated, for instance, by the visibility range of a camera as in Fig. 1.

Problem 2 (*Continuous Min Partition*)**.** For a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ determine a continuous partition $\boldsymbol{\alpha}_2^*$ satisfying

$$\|A\boldsymbol{\alpha}_2^* + \boldsymbol{b}\|_2 = \min_{\underline{\boldsymbol{\alpha}} \le \boldsymbol{\alpha} \le \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2, \tag{8}$$

where A and $\boldsymbol{b}$ are as in (5) and (6), for some constraints vectors $\mathbf{0} \le \underline{\boldsymbol{\alpha}} \le \overline{\boldsymbol{\alpha}} \le \mathbf{1}$.

Observe that the minimization problem (8) is strictly convex, so that it admits a unique minimum. Moreover, the continuous min partition problem (8) has a unique minimizer if and only if the matrix A has a trivial null space or, equivalently, the graph $\mathcal{G}$ is acyclic (Diestel, 2000). We next characterize a relation between the partitioning Problems 1 and 2.