Continuous Graph Partitioning for Camera Network Surveillance

D. Borra [a], F. Pasqualetti [b], F. 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*

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.

Key words: Graph partitioning, constrained optimization, camera network, distributed control.

1 Introduction

Autonomous camera networks are becoming the leading technology for the surveillance of human activities in civil and military applications [2]. 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 [3, 4, 5] 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 [6, 3, 7, 8]. 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 [9, 10, 11].

⋆ 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 [1].

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

In this work we present algorithms for graph partitioning. Our graph partitioning problem differs from classical setups, e.g., see [12, 13, 14, 15]. 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 [16, 17].

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 [18]. 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

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

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 $e = (v_i, v_j) \in \mathcal{E}_{\mathrm{c}}$,

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

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*

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

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

It should be observed that (7) is a convex minimization problem, for which efficient centralized solvers exist [19]. 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*

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

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

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 [20]. We next characterize a relation between the partitioning Problem 1 and 2.

Theorem 1 (Min-max and min partitions) *Let $\boldsymbol{\alpha}_2^*$ be a min partition solution to Problem 2. Then, $\boldsymbol{\alpha}_2^*$ is also a solution to Problem 1, that is,*

$$\|A\boldsymbol{\alpha}_2^* + \boldsymbol{b}\|_\infty = \min_{\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty.$$

In order to prove Theorem 1, we introduce the following definitions and results. For a partition $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ with parameters vector $\boldsymbol{\alpha}$ and length $\mathcal{L}(\mathcal{P}) = \max\{L_1, \dots, L_n\}$, define the *maximal graph* $\mathcal{G}^{\max} = (\mathcal{V}^{\max}, \mathcal{E}^{\max})$, where

$$\mathcal{V}^{\max} = \{v_i \in \mathcal{V} \; : \; L_i = \mathcal{L}(\mathcal{P})\},$$

and $\mathcal{E}^{\max} = (\mathcal{V}^{\max} \times \mathcal{V}^{\max}) \cap \mathcal{E}$.

Lemma 2 (Maximal graph) *Let $\boldsymbol{\alpha}^*$ be a min partition solution to Problem 2, and let $\mathcal{G}^{\max} = (\mathcal{V}^{max}, \mathcal{E}^{max})$ be the maximal graph associated with $\boldsymbol{\alpha}^*$. Then, for all $i \in \mathcal{V}^{max}$ and $j \in \mathcal{V} \setminus \mathcal{V}^{max}$ with $(i,j) \in \mathcal{E}$ it holds*

$$\begin{cases} \alpha_{ij}^* = \underline{\alpha}_{ij}, & \text{if } i < j, \\ \alpha_{ij}^* = \overline{\alpha}_{ij}, & \text{if } i > j, \end{cases}$$

where $0 \leq \underline{\alpha_{ij}} \leq \overline{\alpha_{ij}} \leq 1$ are constraints on α_{ij}.

PROOF. Let $\boldsymbol{\alpha}^*$ be a min partition, let $L^* = A\boldsymbol{\alpha}^* + \boldsymbol{b}$, and let $\mathcal{V}^{\max}$ be the set of vertices satisfying $L_i^* = \|L^*\|_\infty$. Let $(i,j) \in \mathcal{E}$ with $i < j$ and $L_j^* < \|L^*\|_\infty$, that is, $j \in \mathcal{V} \setminus \mathcal{V}^{\max}$. We need to show that $\alpha_{ij}^* = \underline{\alpha}_{ij}$.

Suppose by contradiction that $\alpha_{ij}^* > \underline{\alpha}_{ij}$. Define the partition $\hat{\boldsymbol{\alpha}}$ as $\hat{\alpha}_{hk} = \alpha_{hk}^*$ when $h \neq i$ and $k \neq j$, and $\hat{\alpha}_{ij} = \alpha_{ij}^* - \delta > \underline{\alpha}_{ij}$ for some δ satisfying $\delta \ell_{ij} < (L_i^* - L_j^*)$. Consequently, $\hat{\alpha}_{ji} = 1 - \hat{\alpha}_{ij} = 1 + \delta - \alpha_{ij}^*$. Notice that $\hat{L}_i = L_i^* - \delta \ell_{ij}$ and $\hat{L}_j = L_j^* + \delta \ell_{ij}$. We have

$$\begin{aligned} \|L^*\|_2^2 - \|\hat{L}\|_2^2 &= (L_i^*)^2 - \hat{L}_i^2 + (L_j^*)^2 - \hat{L}_j^2 \\ &= -2\delta^2 \ell_{ij}^2 + 2\delta \ell_{ij} L_i^* - 2\delta \ell_{ij} L_j^* \\ &= 2\delta \ell_{ij} \left(L_i^* - L_j^* - \delta \ell_{ij} \right) > 0, \end{aligned}$$

where the last inequality follows by the choice of δ. Then, the partition $\hat{\boldsymbol{\alpha}}$ achieves a lower cost than $\boldsymbol{\alpha}^*$, which contradicts our assumption of $\boldsymbol{\alpha}^*$ being a min partition. We conclude that $\alpha_{ij}^* = \underline{\alpha}_{ij}$. The case of $\alpha_{ij}^* = \overline{\alpha}_{ij}$ is treated analogously, and the theorem follows. $\square$

We are now ready to prove Theorem 1.

PROOF. Let $\boldsymbol{\alpha}^*$ be a min partition, let $L^* = A\boldsymbol{\alpha}^* + \boldsymbol{b}$, and let $\mathcal{V}^{\max}$ be the set of vertices satisfying $L_i^* = \|L^*\|_\infty$. Assume by contradiction that there exists a partition $\hat{L}$ satisfying $\|\hat{L}\|_\infty < \|L^*\|_\infty$. Then $\hat{L}_i < L_i^*$ for all $i \in \mathcal{V}^{\max}$, and there exists at least one vertex $j \in \mathcal{V} \setminus \mathcal{V}^{\max}$, with $(i,j) \in \mathcal{E}$ for some $i \in \mathcal{V}^{\max}$, satisfying $\hat{L}_j > L_j^*$. In other words, the load removed from the vertices in $\mathcal{V}^{\max}$ must be sustained by neighboring vertices in $\mathcal{V} \setminus \mathcal{V}^{\max}$. This statement contradicts Lemma 2, and the claimed statement follows. $\square$

Theorem 1 implies that a solution to the non-differentiable Problem 1 can be computed by solving the differentiable Problem 2. Hence, the distributed procedure developed in [21] can be used to compute an optimal min-max partition solution to Problem 1 (see Section 5). In the next sections we exploit the continuous graph partitioning problem to design of cameras trajectories.

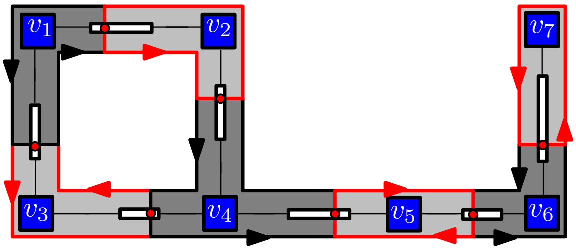

Fig. 1. This figure shows an environment surveilled by a camera network. Cameras are installed at the locations $\mathcal{V}_c = \{v_1, \dots, v_7\}$. White rectangles along the edges represent cameras visibility constraints, and the parameters $\boldsymbol{\alpha}$ define a continuous partition of $\mathcal{G}$. Finally, the DF-Trajectory associated with the partition given by $\boldsymbol{\alpha}$ is identified by the closed paths around the cameras.

Remark 1 (Unconstrained partitions) *Consider the unconstrained partitioning Problem 2 ($\underline{\alpha} = -\infty$ and $\overline{\alpha} = \infty$). Observe that, since the sum of all edges length is constant and independent of the partition, the solution to Problem 2 is obtained by enforcing an equal load for each vertex. Define the consensus vector*

$$L^* = \left(\sum_{(i,j)\in\mathcal{E}} \frac{\ell_{ij}}{n} \right) \boldsymbol{1},$$

and notice that every minimizer to Problem 2, and in fact to Problem 1 as well, can be written as

$$\boldsymbol{\alpha}^* = A^\dagger (L^* - \boldsymbol{b}) + \boldsymbol{w} = -A^\dagger \boldsymbol{b} + \boldsymbol{w},$$

where $A^\dagger$ is the Moore-Penrose pseudoinverse of A, $A\boldsymbol{w} = \boldsymbol{0}$, and $A^\dagger L^* = A^\mathsf{T} L^* = 0$ due to the definitions of the incidence matrix A [20]. Additionally, if $\boldsymbol{\alpha}^*$ satisfies $\boldsymbol{0} \le \boldsymbol{\alpha}^* \le \boldsymbol{1}$, then $\boldsymbol{\alpha}^*$ is a solution to the constrained Problem 1 and 2. □

3 Setup for Camera Surveillance

In this section we describe our setup for cameras surveillance and we introduce some preliminary notions. We consider the problem of surveilling an environment by means of a network of PTZ cameras installed at fixed locations. In particular, we let cameras $\mathcal{V} = \{1, \dots, n\}$ be installed at the locations $\{v_1, \dots, v_n\}$, and we define the graph (roadmap) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $(i,j) \in \mathcal{E}$ whenever the segment $[v_i, v_j]$ belongs to the environment (locations v_i and v_j are within line of sight). Let the weight of the edge $(i,j) \in \mathcal{E}$ equal the length $\ell_{ij} = \|v_i - v_j\|_2$, and define

$$\ell^{\max} = \max\{\ell_{ij} \ : \ (i,j) \in \mathcal{E}\}.$$

To simplify notation, we let cameras be installed at every vertex of $\mathcal{G}$. Our results are however more general, and extend directly to the case where cameras are installed only at a subset of vertices covering the whole graph with their f.o.v..

Let $x_i(t)$ denote the position at time t of the f.o.v. of the i-th camera. We assume that each camera has a limited visibility range along each adjacent edge. In particular,

(A1) $x_i(t) \in [v_i, \ v_j]$ at all times $t \in \mathbb{R}_{\ge 0}$ for some $j \in \mathcal{N}_i$;

(A2) the i-th f.o.v. moves at unit or zero speed.

Our setup is illustrated in Fig. 1. [1]

A *cameras trajectory* is a set of n continuous functions $X = \{x_1, \dots, x_n\}$, where $x_i : \mathbb{R}_{\ge 0} \to \cup_{(i,j)\in\mathcal{E}} [v_i, v_j]$ describes the position of the i-th f.o.v. along the roadmap $\mathcal{G}$. We focus on *periodic* cameras trajectories, for which there exists a finite time $T \in \mathbb{R}_{\ge 0}$ satisfying $X(t+T) = X(t)$ for all $t \in \mathbb{R}_{\ge 0}$. Define the *image* of the i-th camera as the set of points visited by the i-th f.o.v. in any period of length T, that is,

$$\mathrm{Im}(x_i) = \cup_{t\in[0,T]}\, x_i(t),$$

and the cameras *image set* as $\mathcal{I}^X = \{\mathrm{Im}(x_1), \dots, \mathrm{Im}(x_n)\}$.

We allow for the presence of intruders along the roadmap and we consider the design of camera trajectories for the quickest detection of intruders. We let intruders appear at arbitrary locations and times, and we assume that an intruder is detected as soon as the f.o.v. of a camera coincides with the position of the intruder.

The trajectory of an intruder is a continuous function $p : \mathbb{R}_{\ge 0} \to \cup_{(i,j)\in\mathcal{E}} [v_i, v_j]$. We focus on static intruders, where $p(t) = p_0$ for all $t \ge t_0$ and for some $p_0 \in \mathcal{E}$. We define the *worst-case detection time* of a cameras trajectory as the longest time for the detection of an intruder. In particular, for an intruder appearing at time t_0 and location p_0, and a cameras trajectory X, let [2]

$$t^*(t_0, p_0, X) = \min\{t - t_0 \ : \ t > t_0,\, p_0 \in X(t)\}.$$

We define the worst-case detection time as

$$\begin{aligned} \mathrm{WDT}(X) = \sup_{t_0, p_0} \ & t^*(t_0, p_0, X), \text{ with} \\ & p_0 \in [v_i, v_j] \text{ and } (i,j) \in \mathcal{E}, \\ & t_0 \in [0, T]. \end{aligned} \tag{9}$$

To conclude this section, notice that $\mathrm{WDT}(X) < \infty$ if and only if the entire roadmap is persistently surveilled by the cameras, that is, $\mathcal{E} \subseteq \mathrm{Im}(X)$.

---

[1] Our assumption of point-wise f.o.v. does not prevent applicability of our results to real scenarios, while being convenient for the analysis. See also [22] for a detailed discussion of our assumptions and for experimental results.

[2] If $p_0 \notin X(t)$ for all $t \in \mathbb{R}_{\ge t_0}$, then $t^*(t_0, p_0, X) = \infty$.

Trajectory 1 DF-Trajectory for i-th camera

Input: Parameters α_{ij} and set of neighbors $\mathcal{N}_i$;

Let $s_i(t) = (tv_{ij} + (\alpha_{ij}\ell_{ij} - t)v_i)/\alpha_{ij}\ell_{ij}$, for $t \in [0, \alpha_{ij}\ell_{ij}]$;
Set $t_0 = 0$;
for $v_j \in \mathcal{N}_i$ **do**
  Set v_{ij} as in Eq. (2);
  $x_i(t) = s_i(t - t_0)$, for $t \in [t_0, t_0 + \alpha_{ij}\ell_{ij}]$;
  $x_i(t) = s_i(2\alpha_{ij}\ell_{ij} - (t - t_0))$, for $t \in [t_0 + \alpha_{ij}\ell_{ij}, t_0 + 2\alpha_{ij}\ell_{ij}]$;
  $t_0 = t_0 + 2\alpha_{ij}\ell_{ij}$;
end for

4 Camera Trajectory for Intruder Detection

We now define a particular cameras trajectory. Let $\boldsymbol{\alpha}^{\text{df}}$ define the continuous partition $\mathcal{P}^{\text{df}}$ as in (1). The *DF-Trajectory* X^{df} with image set $\mathcal{P}^{\text{df}}$ is obtained by letting each camera sweep its subroadmap in, for instance, depth-first order [20]. The DF-Trajectory is formally described in Trajectory 1, and illustrated in Fig. 1. We next show that cameras trajectories designed from a continuous roadmap partition achieve detection performance within a constant factor of optimal for general networks.

Theorem 3 (Worst-case detection time for DF-Trajectory) *For the roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\mathcal{P}^*$ be a min partition solution to Problem 2. Let X^* be the DF-Trajectory with image set $\mathcal{P}^*$. Then,*

(i) $\mathrm{WDT}(X^*) = 2\mathcal{L}(\mathcal{P}^*)$ *, and*

(ii) $\mathrm{WDT}(X^*) \leq 2\,\mathrm{WDT}^*$.

PROOF. Statement (i) follows from the definition of DF-Trajectory, because each camera sweeps its assigned subroadmap at maximum speed along a depth-first tour.

To show statement (ii) consider a min partition $\boldsymbol{\alpha}$, and let $\mathcal{G}^{\max} = (\mathcal{V}^{\max}, \mathcal{E}^{\max})$ be its associated maximal graph. Define $\mathrm{Length}(\mathcal{G}^{\max}) = \sum_{(i,j)\in\mathcal{E}^{\max}} \|v_i - v_j\|_2$, and notice that

$$\mathrm{WDT}^*_{|\mathcal{G}^{\max}} \geq \frac{\mathrm{Length}(\mathcal{G}^{\max})}{|\mathcal{V}^{\max}|}, \tag{10}$$

where $\mathrm{WDT}^*_{|\mathcal{G}^{\max}}$ denotes the smallest worst-case detection time for $\mathcal{G}^{\max}$. Indeed, since $L_i = L_j$ for each $i, j \in \mathcal{V}^{\max}$, each camera needs to sweep (at unit speed) an path of length $\frac{\mathrm{Length}(\mathcal{G}^{\max})}{|\mathcal{V}^{\max}|}$ for $\mathcal{G}^{\max}$ to be covered. Moreover, due to Lemma 2, cameras outside $\mathcal{G}^{\max}$ cannot visit any point in the interior of $\mathcal{G}^{\max}$. We have

$$\mathrm{WDT}^* \geq \mathrm{WDT}^*_{|\mathcal{G}^{\max}} \geq \frac{\mathrm{Length}(\mathcal{G}^{\max})}{|\mathcal{V}^{\max}|} = \frac{\mathrm{WDT}(X^*)}{2},$$

where the first inequality follows because WDT^* is defined as the maximum detection time over all points along $\mathcal{G}$, and not only $\mathcal{G}^{\max}$, the second inequality follows from (10), and the last equality from statement (i) and (3). This concludes the proof of statement (ii).

Algorithm 2 Asymmetric Broadcast Partitioning

Define S_i^t as in equation (11);
Camera i is randomly selected;
Receive S_j^t from all cameras $j \in \mathcal{N}_i^{\text{in}}$, and perform the following operations (other cameras perform no operations);
for $j \in \mathcal{N}_i$ **do**
  $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon\ell_{ij}(L_i^t - L_j^t)$;
  if $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$ **then** $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$;
  else if $\alpha_{ij}^{t+1} > \overline{\alpha}_{ij}$ **then** $\alpha_{ij}^{t+1} = \overline{\alpha}_{ij}$;
  end if
end for
Camera i transmits S_i^{t+1} to camera j, for all $j \in \mathcal{N}_i$.

In Theorem 3 we show that the continuous graph partitioning problem can be effectively used to design constant factor optimal cameras trajectories. It can be shown that optimality is in fact achieved for tree and ring roadmaps, and for other particular topologies.

5 A Distributed Partitioning Algorithm

In this section we design a distributed algorithm for the continuous min-max partitioning problem. Given an optimal partition, cameras organize along a DF-Trajectory as in Trajectory 1. We assume each camera to be equipped with a wireless sensor device. Let

$$S_i^t = \{\alpha_{ij}^t \ : \ j \in \mathcal{N}_i\} \tag{11}$$

be the state of camera i at iteration $t \in \mathbb{N}$. Finally, let $\alpha_{ij}^0 = \underline{\alpha}_{ij}$ for all $(i,j) \in \mathcal{E}$ with $i < j$.

We assume an asymmetric broadcast communication protocol. In particular, at each iteration only one camera updates its state by using local information from its neighboring cameras. In order to guarantee the convergence of the algorithm, we assume the existence of a finite duration $\tau \in \mathbb{R}_{>0}$ such that, for all $t \in \mathbb{R}_{\geq 0}$, every camera is selected at least once in the time interval $[t, t+\tau)$ (*partial asynchronism assumption* [21]). The t-th iteration of this algorithm is detailed in Algorithm 2.

Theorem 4 ***(Asymmetric Broadcast Partitioning)*** *For the roadmap $\mathcal{G}$, let A and $\boldsymbol{b}$ be as in* (5) *and* (6), *respectively. Let τ be the partial asynchronism constant, and let $0 < \varepsilon < (K(1 + \tau + \tau|\mathcal{E}_{\text{c}}|))^{-1}$, where $K \in \mathbb{R}_{>0}$ is the Lipschitz constant of $\boldsymbol{\alpha} \to A^T(A\boldsymbol{\alpha} + \boldsymbol{b})$. Then, the* Asymmetric Broadcast Partitioning *algorithm in Algorithm 2 asymptotically converges to $\boldsymbol{\alpha}^*_{\text{AB}} = \lim_{t\to\infty} \boldsymbol{\alpha}^t$. Moreover,*

$$\min_{\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}^*_{\text{AB}} + \boldsymbol{b}\|_\infty^2,$$

where $\underline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\alpha}}$ denote the cameras constraints.

PROOF. The algorithm update follows the gradient of $\boldsymbol{\alpha} \to \frac{1}{2}\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2$. Because of the partial asynchronism assumption and the fact that $\boldsymbol{\alpha}^t \in [0,1]^{\mathcal{E}_c}$ is such that $\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha}^t \leq \overline{\boldsymbol{\alpha}}$ for all $t \in \mathbb{N}$, the statement follows from [21, Section 7, Proposition 5.3] and Theorem 1. Note that the bound for the stepsize ε depends on the Lipschitz constant K, the time horizon τ and the number of edges connecting the cameras [21]. $\square$

6 Conclusion

In the first part of this note we introduce and solve the continuous graph partitioning problem, where each edge is partitioned in a continuous fashion among its endpoint vertices, and where the cost function is given by the largest load of each vertex. In the second part of the paper we show how the continuous graph partitioning problem can be used to design surveillance trajectories for a network of autonomous PTZ cameras for the quickest detection of intruders. We only consider the case of static intruders and we leave the case of dynamic intruders as the subject of future research. We conclude with a distributed algorithm for the design of cameras trajectories based on continuous graph partitioning.

References

[1] D. Borra, F. Pasqualetti, and F. Bullo. Continuous graph partitioning for camera network surveillance. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 228–233, Santa Barbara, CA, USA, September 2012.

[2] Eds. B. Bhanu, C. Ravishankar, A. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos. *Distributed Video Sensor Networks.* Springer, 2010.

[3] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato. Distributed perimeter patrolling and tracking for camera networks. In *IEEE Conf. on Decision and Control*, pages 2093–2098, Atlanta, GA, USA, December 2010.

[4] R. Carli, A. Cenedese, and L. Schenato. Distributed partitioning strategies for perimeter patrolling. In *American Control Conference*, pages 4026–4031, San Francisco, CA, USA, June 2011.

[5] M. Spindler, F. Pasqualetti, and F. Bullo. Distributed multi-camera synchronization for smart-intruder detection. In *American Control Conference*, pages 5120–5125, Montréal, Canada, June 2012.

[6] R. Alberton, R. Carli, A. Cenedese, and L. Schenato. Multi-agent perimeter patrolling subject to mobility constraints. In *American Control Conference*, pages 4498–4503, Montréal, Canada, June 2012.

[7] A. Machado, G. Ramalho, J. D. Zucker, and A. Drogoul. Multi-agent patrolling: An empirical analysis of alternative architectures. In *Multi-Agent-Based Simulation II*, Lecture Notes in Computer Science, pages 155–170. Springer, 2003.

[8] F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms. *IEEE Transactions on Robotics*, 28(3):592–606, 2012.

[9] A. Kehagias, G. Hollinger, and S. Singh. A graph search algorithm for indoor pursuit/evasion. *Mathematical and Computer Modelling*, 50(9-10):1305–1317, 2009.

[10] A. Kolling and S. Carpin. Pursuit-evasion on trees by robot teams. *IEEE Transactions on Robotics*, 26(1):32–47, 2010.

[11] T. D. Parsons. Pursuit-evasion in a graph. In Y. Alavi and D. Lick, editors, *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. Springer, 1978.

[12] K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.

[13] E. M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.

[14] G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha. Min–max tree covers of graphs. *Operations Research Letters*, 32(4):309–315, 2004.

[15] G. Even, J. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 639–648, New Orleans, LA, USA, January 1997.

[16] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279–301, 1989.

[17] R. Lüling, B. Monien, and F. Ramme. Load balancing in large networks: A comparative study. In *IEEE Symposium on Parallel and Distributed Processing*, pages 686–689, Dallas, TX, USA, December 1991.

[18] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, 2009.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[20] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer, 2 edition, 2000.

[21] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Athena Scientific, 1997.

[22] F. Pasqualetti, F. Zanella, J. R. Peters, M. Spindler, R. Carli, and F. Bullo. Camera network coordination for intruder detection. *IEEE Transactions on Control Systems Technology*, 22(5):1669–1683, 2014.