

Loss of Distributed Coverage Using Lazy Agents Operating Under Discrete, Local, Event-Triggered Communication

Edward Vickery* and Aditya A. Paranjape†

Abstract

Continuous surveillance of a spatial region using distributed robots and sensors is a well-studied application in the area of multi-agent systems. This paper investigates a practically-relevant scenario where robotic sensors are introduced asynchronously and inter-robot communication is discrete, event-driven, local and asynchronous. Furthermore, we work with lazy robots; i.e., the robots seek to minimize their area of responsibility by equipartitioning the domain to be covered. We adapt a well-known algorithm which is practicable and known to generally work well for coverage problems. For a specially chosen geometry of the spatial domain, we show that there exists a non-trivial sequence of inter-robot communication events which leads to an instantaneous loss of coverage when the number of robots exceeds a certain threshold. The same sequence of events preserves coverage and, further, leads to an equipartition of the domain when the number of robots is smaller than the threshold. This result demonstrates that coverage guarantees for a given algorithm might be sensitive to the number of robots and, therefore, may not scale in obvious ways. It also suggests that when such algorithms are to be verified and validated prior to field deployment, the number of robots or sensors used in test scenarios should match that deployed on the field.

1 Introduction

The development of autonomous vehicles in recent years has expanded the range of tasks that can be carried out without human intervention. This paper explores one such complex task, namely that of continuous surveillance of an environment using autonomous mobile robots. The problem of surveillance and monitoring has multiple facets, depending on the nature of the mission and the sensors and robots involved in the task [6]. The problem addressed in the paper is effectively that of partitioning an environment for the purpose of continuous surveillance using a distributed scheme which relies on local, event-triggered communication between mobile robots. The robots seek to minimize their individual areas of coverage while ensuring that the environment as a whole is covered. The distributed nature of the communication and task allocation between the robots leads to the following question with practical ramifications: *if a reasonably designed coverage algorithm is proven to work for some non-trivial range of numbers of robots, can it fail to work when the number of robots is changed to outside the proven range?* We answer this question in the affirmative by constructing an example.

1.1 Overview of the literature

Optimal sensor placement problems are built upon the premise that the number of sensors and their placement can be mapped to an objective function which needs to be either maximized or minimized subject

*EV is Head of Solution Operations at SHL, London, United Kingdom. Email: ted.vickery@gmail.com

†AP is Senior Scientist at TCS Research, Tata Consultancy Services Ltd., Pune. He is also Honorary Lecturer at Imperial College London and Visiting Associate Professor at the Indian Institute of Technology Bombay. Email: aditya.paranjape@gmail.com.

to constraints related to the sensor and the environment. When the objective function satisfies submodularity properties, it is possible to use greedy algorithms to solve these problems up to provable bounds [7]. This approach has been investigated for sensor placement in [13, 12, 14]. Similar work in [20] presents an efficient path planning algorithm for multiple agents in the presence of resource constraints for the agents. Integer programming techniques can be employed if the structure of the problem permits appropriate spatial discretization [5].

All of these approaches assume the existence of a centralized decision-making system that is aware of all active sensors and can allocate them to individual positions. The centralized decision-making approach becomes impractical when robotic agents are inserted into environments where continuous communication with a central node is impossible (e.g., hostile environments or ones which are naturally packed with communication obstacles). In such cases, inter-robot communication is asynchronous, local and event-driven. A canonical coverage problem is that of partitioning the environment between the mobile robots equitably and dynamically using a distributed scheme.

Distributed coverage algorithms that achieve a Voronoi partition of convex environments using a distributed approach were proposed in [8]. The distributed scheme is dynamic, in that the Voronoi partitions are scaled and rearranged dynamically to yield an optimal partition together with a motion planning algorithm for the mobile robots. This approach has been extended to non-convex domains [17, 3, 2], to robots with finite communication radii [9], to environments with unknown sensory functions [18] and to problems where the cost function for a sensor can be augmented or mixed with that of its neighbors [19]. A similar approach can be used to balance raw coverage (i.e., the area covered) with the quality of the coverage for a spatial distribution of events [1].

The distinction between continuous coverage and reliable detection of flag events (for which continuous coverage is sufficient but not necessary) is brought out for time varying environments in [15] wherein sensor movement is optimized in order to maximise the probability of identifying flag events.

A continuous flow of information may not be required in order to maintain coverage. An algorithm which uses events triggered by individual agents in order to guarantee coverage is presented in [16]. An algorithm which caters to gossip-based inter-robot communication has been investigated in [10] where random pairs of agents are allowed to communicate and relocate based on local information exchange.

1.2 Contribution

The algorithm analyzed in this paper is an adaptation of the algorithm in [10] for robots that use a lazy scheme (in a sense which will be made precise later) to repartition or resize their areas of responsibility. We wish to examine whether such lazy behavior (which may be viewed as the equivalent of greedy behavior in optimization problems with an agent-level penalty function that dominates the global reward function for coverage) can result in a loss of coverage and the conditions under which coverage is lost.

Towards that end, we construct a simplified example and a sequence of events which leads to an instantaneous loss of coverage when the number of robots exceeds a non-trivial threshold. Interestingly, the same sequence of events actually leads to an equipartition of the domain (i.e., the optimum solution) for a smaller number of robots. This demonstration suggests that the success of multi-agent algorithms operating in the presence of restricted communication might be sensitive to the number of agents involved, above and beyond the known complexities that arise due to the “scale” of the problem or the geometry of the environment.

The rest of the paper is organized as follows. Preliminaries are laid out in Sec. 2, including the class coverage algorithms considered in the paper. The main theoretical results of the paper are presented in Sec. 3, and numerical experiments are used to generalize those results in Sec. 4.

2 Preliminaries

2.1 Unit circle

Let S^1 denote the unit circle parametrized by the angular variable $\theta \in [0, 2\pi)$. We write $\theta \in S^1$. Let $\theta_1, \theta_2 \in S^1$. The length of the shorter arc between these points is given by

$$d(\theta_1, \theta_2) = d(\theta_2, \theta_1) = \begin{cases} |\theta_2 - \theta_1|, & |\theta_2 - \theta_1| < \pi \\ 2\pi - |\theta_2 - \theta_1| & \text{otherwise} \end{cases} \quad (1)$$

The centroid of the two points along the short arc is given by

$$\text{mid}(\theta_1, \theta_2) = \begin{cases} 0.5(\theta_2 + \theta_1), & |\theta_2 - \theta_1| < \pi \\ \text{mod}(\pi + 0.5(\theta_2 + \theta_1), 2\pi) & \text{otherwise} \end{cases} \quad (2)$$

The set S^1 is isomorphic to a unit circle in the complex plane via the invertible mapping $T(\theta) = e^{j\theta}$, $\theta \in S^1$.

Definition 1 (Positive or clockwise rotation). *An arc in S^1 is said to be a clockwise or positively directed arc from $\theta_l \in S^1$ to $\theta_u \in S^1$ if there exists a continuous function $c : [0, 1] \rightarrow \mathbb{R}$ and $\omega \geq 0$ such that: (i) every point on the arc can be represented as $c(t)$ for some $t \in [0, 1]$, with $c(0) = \theta_l$ and $c(1) = \theta_u$; (ii) $T(c(t)) = e^{j\omega t}T(c(0))$; and (iii) $c(t_1) = c(t_2)$ for $t_1 \neq t_2$ iff $\theta_l = \theta_u$ and $\omega = 0$.*

Definition 2. *We say that $\theta_u \succ \theta_l$ if the shortest arc from θ_l to θ_u (in that order) is traced via a clockwise rotation.*

Definition 3. *We define the addition operator \oplus to denote a clockwise rotation on S^1 ; i.e., $\theta_1 \oplus \theta_2$ denotes a clockwise rotation of magnitude θ_2 starting from θ_1 . It is clear that $\theta_1 \oplus \theta_2 = \theta_2 \oplus \theta_1$. We define the operator \ominus to denote an anticlockwise rotation on S^1 : $\theta_1 \ominus \theta_2$ denotes an anti-clockwise rotation of magnitude θ_2 starting with θ_1 . Finally, the standard summation operator $\sum(\cdot)$ will denote a sum using \oplus when the arguments belong to S^1 .*

If $\theta_2 \succ \theta_1$, Definition 3 allows us to write

$$\theta_1 = \text{mid}(\theta_1, \theta_2) \ominus 0.5 d(\theta_1, \theta_2), \quad \theta_2 = \text{mid}(\theta_1, \theta_2) \oplus 0.5 d(\theta_1, \theta_2), \quad (3)$$

2.2 Partitions of closed, bounded regions in \mathbb{R}^2

Let $Q \subset \mathbb{R}^2$ be a closed, bounded domain containing $N \geq 1$ agents or sensors. Let $p_i \in Q$ denote the position of the i^{th} agent. A partition of size N of Q is a set $V = \{V_1, \dots, V_N\}$, where $V_i \subset Q$ are closed and satisfy $\cup_{i=1}^N V_i = Q$. We may further prescribe that each agent lie inside its own partition. The coverage problem usually considered in the literature involves finding a partition V^* which solves the problem

$$V^* = \arg \min_V \mathcal{H}_V(\mathcal{P}) \triangleq \sum_{i=1}^n \int_{V_i} f_i(\|q - p_i\|) d\phi(q) \quad (4)$$

where $\phi : Q \rightarrow \mathbb{R}_+$ satisfying $\int_Q d\phi(q) = 1$ is a weighting function (also called the sensing function) and $f_i : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ represents the sensing performance of the agent i as a function of its distance from the sensed location $q \in Q$. Lloyd's algorithm and its variants [4] are used to partition Q dynamically into Voronoi cells which leads, in turn, to the optimal partition which is itself a Voronoi partition. Partitions may also be constructed organically to satisfy sensing constraints, such as in [11], without solving the optimization problem above. In this paper, we will consider equipartitions which are defined as follows.

Definition 4. For N agents located at $\{p_1, \dots, p_N\}$, $p_i \in Q$ for all i , we call $W = \{W_1, \dots, W_N\}$ an equipartition of Q if it satisfies: (i) $W_i \subseteq Q$ is closed for all i ; (ii) $p_i \in W_i$; (iii) $\text{area}(W_i) = \text{area}(W_j) = \text{area}(Q)/N$ for all i, j ; and (iv) $\cup_{i=1}^N W_i = Q$. It follows that $\text{int}(W_i) \cap \text{int}(W_j) = \emptyset$ for $i \neq j$ and $p_i \in \text{int}(W_j)$ if and only if $i = j$.

It is clear that an equipartition W solves (4) when $\phi(q)$ is uniform and the function $f_i(\cdot)$ is identical for all i .

2.3 Gossip-based coverage algorithms

In this section, we describe the *class of* distributed coverage algorithms, based on [10], which are subsequently specialized and analyzed in the paper. Briefly, a sufficient condition for coverage is that the spatial domain equals the union of the areas assigned to individual agents. This assignment is carried out in a distributed manner by the agents with the aim of creating an equipartition while ensuring that coverage is not lost. We make four important assumptions: (1) the agents are assumed to be *lazy* in a sense that will be made precise presently; (2) every agent is aware of the geometry of the environment before entry but not of the other agents; (3) at most one *interaction event* (defined presently) can occur at any given point in time, and (4) agents' response to events, including repartitioning, is instantaneous and is thus complete before the next event. Unlike [16], we assume that an interaction event cannot be triggered by one or more agents. Rather, we model it as a random occurrence, which is a proxy for two agents coming within their mutual communication radius either in the course of exploring their area of responsibility or responding to an environmental event.

Definition 5. Consider an agent in a closed, connected domain Q with area $|Q|$ and suppose that it has knowledge K about the agents in Q , with $|K|$ equal to the number of agents that it is aware of (including itself). Suppose that the agent allocates itself a domain $A \subseteq Q$ with area $|A|$. The agent is said to be *lazy* with an ϵ degree of altruism if

$$|A| = \frac{|Q|}{|K|} + \epsilon$$

where ϵ can be time-varying and agent-specific. The agent is said to be *lazy* (with no mention of altruism) if $\epsilon = 0$.

Remark 1. The numerical area $|A|$ of a partition A can be calculated by scaling using the sensor function $\phi(q)$ ($q \in Q$) to ensure that the partitions are equitable. We assume that $\phi(\cdot)$ is uniform, so that $|A|$ is the usual Euclidean area of A .

Definition 6. An interaction event, identified by the time t at which it occurs, is defined as an interaction between precisely two agents i and j . The interaction consists of (i) updating each agent's knowledge $K_i, K_j \leftarrow K_i \cup K_j$, and (ii) repartitioning and resizing of their individual areas of responsibility as per the guiding algorithm. Since we do not model environmental events in this paper, we will use the word *event* to refer hereafter to an interaction event.

There are two types of (interaction) events. In the first type, an agent enters the domain for the first time, with no knowledge of the other agents, and encounters an existing agent. It makes a copy of the knowledge possessed by the existing agent and the two agents partition *the existing agent's* area equitably into disjoint halves while also taking over responsibility for some of the surrounding area. In the second type of events, an agent that is already in the domain interacts with another agent that is also in the domain. The two agents update each other's information so that both possess the union of their prior individual information. They repartition their individual areas as per the problem-specific guiding algorithm and the process continues. The pseudocode is described in Algorithm 1.

We note that several steps of Algorithm 1 have been deliberately left vague. These can be made precise for individual problems, as we illustrate in the next section. We have only prescribed that (i) exchange of knowledge should correspond to all agents learning the union of their prior knowledge, and (ii) the areas assigned or reassigned to all participants in an interaction should have equal magnitudes up to the degree of altruism of individual agents.

Algorithm 1 General algorithm for partitioning a domain

Require: A domain $Q \in \mathbb{R}^2$ and N agents in all

Initialize: time $t = 0$; initial positions and areas of responsibility for the agents already in Q . The union of the areas of responsibility of agents in Q equals Q

while Stopping condition not reached **do**

if Enters new agent j **then**

 New agent j interacts with agent i already in Q

 Knowledge exchange $K_j \leftarrow K_i$

 Disjoint areas of responsibility assigned $A_i, A_j \in Q$; $|A_i| = |A_j| \sim 1/|K_i| + \epsilon[t]$.

else

 Existing agents i, j interact

 Knowledge sharing: $K_i = K_j \leftarrow K_i \cup K_j$

 Areas of responsibility reassigned so that $|A_{\{i,j\}}| \sim Q/|K_{\{i,j\}}| + \epsilon_{\{i,j\}}[t]$

end if

 Stopping condition reached if $t = t_{\max}$ or Q is equipartitioned or no further events are feasible

end while

3 Main Results

In this section, we consider a simplified coverage problem and apply a corresponding manifestation of Algorithm 1 to solve it. We assume that the agents are lazy (i.e., $\epsilon = 0$ uniformly for all agents). We construct a sequence of events in Sec. 3.3 which leads to equipartition when the number of agents is less than a certain threshold, and to loss of coverage when the number of agents exceeds the threshold.

3.1 Geometry of the domain and some notation

Let $Q_A = \{(x, y) \in \mathbb{R}^2 \mid \delta^2 < x^2 + y^2 \leq 1\}$ denote the domain of interest in \mathbb{R}^2 , where $0 < \delta \ll 1$. Clearly, Q_A is a unit disc with a small hole at the centre. We can recast the problem into one of partitioning the domain $Q = S^1$ (the unit circle). It is evident from basic geometry that a partition of S^1 can be mapped to an equivalent angular slice of Q_A . Thus, an equipartition of S^1 can be mapped to an equipartition of Q_A .

The angular position of the i^{th} agent in $Q = S^1$ is denoted by $\theta_i \in [0, 2\pi)$ once it is introduced in the domain, and $n_i \geq 1$ denotes the number of agents that it is aware of, including itself. The arc of dominance of the agent i is denoted by $[\theta_i^l, \theta_i^u]$ with $\theta_i^u \succ \theta_i^l$ and $\theta_i^c = \text{mid}(\theta_i^l, \theta_i^u)$. Finally, let $S_i = d(\theta_i^l, \theta_i^u)$. We note that all of these variables are functions of time t ; for brevity, we omit that argument unless necessary.

Definition 7. *An agent is said to be lazy if its arc of dominance on S^1 has length $S_i = 2\pi/n_i$.*

The following result can be derived readily using basic geometry and we omit a formal proof.

Lemma 1. *Suppose two lazy agents i and j are aware of n_i and n_j agents, respectively. Then, the overlap between their areas of dominance $S_{i,j} > 0$ if and only if*

$$\frac{\pi}{n_i} + \frac{\pi}{n_j} - d(\theta_i^c, \theta_j^c) > 0$$

Moreover, if $n_i > 1$ and $n_j > 1$, the overlap is given by

$$S_{i,j} = \min \left(\max \left(\frac{\pi}{n_i} + \frac{\pi}{n_j} - d(\theta_i^c, \theta_j^c), 0 \right), \frac{2\pi}{n_i}, \frac{2\pi}{n_j} \right) \quad (5)$$

If $n_i = 1$, $S_{i,j} = 2\pi/n_j$, and likewise if $n_j = 1$.

Definition 8. For every agent j , we denote the area of overlap with its immediate clockwise and anti-clockwise neighbors by C_j^u and C_j^l , respectively. In terms of the notation introduced in Lemma 1,

$$\begin{aligned} C_j^u &= S_{j,k}, k = \arg \min_{i \neq j} \{d(\theta_j^c, \theta_i^c) \mid \theta_i^c \succ \theta_j^c\} \\ C_j^l &= S_{j,k}, k = \arg \min_{i \neq j} \{d(\theta_j^c, \theta_i^c) \mid \theta_i^c \prec \theta_j^c\} \end{aligned}$$

3.2 Addition of a new agent and repartitioning

Suppose that a new agent k is added to Q at time t and it interacts with an agent j that is already in Q . As per Algorithm 1, they update their knowledge of the number of agents so that $n_k[t] = n_j[t] = n_j[t-1] + 1$. Thereafter, the two agents j and k assign themselves an area of responsibility of size $2\pi/n_k[t]$ such that the two arcs intersect at the point $\theta_j^c[t-1]$ (i.e., at the midpoint of agent j 's previous area of responsibility). The pseudocode for this process is presented in Algorithm 2.

Algorithm 2 New agent repartition algorithm

Require: agent i in Q located at θ_i^c and aware of $n_i (\geq 1)$ agents

Require: agent j enters Q and interacts with i

$$\begin{aligned} n_j &\leftarrow n_i + 1, n_i \leftarrow n_i + 1 \\ \theta_i^u &\leftarrow \theta_i^c, \theta_j^l \leftarrow \theta_i^c \\ \theta_i^l &\leftarrow \theta_i^u \ominus 2\pi/n_i, \theta_i^c \leftarrow \text{mid}(\theta_i^l, \theta_i^u) \\ \theta_j^u &\leftarrow \theta_j^l \oplus 2\pi/n_j, \theta_j^c \leftarrow \text{mid}(\theta_j^l, \theta_j^u) \end{aligned}$$

The next result shows that the process of adding a new agent is not detrimental to instantaneous coverage in itself.

Lemma 2. Suppose that the domain $Q = S^1$ is covered by $k > 1$ agents and suppose that a new agent $k+1$ is added to the domain Q . Then, Algorithm 2 ensures that instantaneous coverage is not lost. Moreover, the resulting area of dominance of agent $k+1$ overlaps with that of at least one other agent in $[1, k]$

Proof. When the $(k+1)^{\text{th}}$ is introduced, let i denote the index of the agent with whom it interacts. These two agents update their knowledge of the number of agents in Q to $n_i + 1$ as per Algorithm 2, where n_i is the number of agents known to the agent i before interacting with agent $k+1$. The two agents also assign themselves disjoint partitions of size $S_i = S_{k+1} = 2\pi/(n_i + 1)$ each, which yields a joint coverage of size $4\pi/(n_i + 1)$. We note that

$$4\pi/(n_i + 1) > 2\pi/n_i \text{ for } n_i > 1 \quad (6)$$

Algorithm 2 ensures that the areas of dominance of agents i and $k+1$ are symmetric about the centroid of the previous area of dominance (labeled temporarily as S_i^{old}) of agent i . From (6), it follows that the combined area of dominance of agents i and $k+1$ is a superset of S_i^{old} . Since Q was covered fully by the k agents before the entry of agent $k+1$, agents other than i and $k+1$ cover the area outside S_i^{old} . This completes the proof. \square

Lemma 2 implies that (i) the domain Q is fully covered, with redundant coverage in some areas, at each step of agent addition, and (ii) the partition of Q is not an equipartition at the end of agent addition. Since the agents would ideally aim for an equipartition, that would require further interaction between the agents.

3.3 Sequential interaction of lazy agents

In this section, we construct the sequence of interactions between agents. The sequence has two components. First, the agents are introduced sequentially as described in Algorithm 3. The sequence continues with inter-agent interaction chosen from one of Algorithm 4 and Algorithm 5.

3.3.1 Addition of agents

In the first step (Algorithm 3), agents are introduced sequentially with further prescription that agent $j > 1$ interacts only with agent $j - 1$ upon entering the domain. This continues until the last agent N is introduced.

Algorithm 3 Special case: sequential addition and interaction

```

Initialize: domain  $Q = S^1$ ; number of agents  $n > 2$ 
Initialize: agents_added = 2;  $t = 2$ ;  $n_1 = n_2 = 2$ 
Initialize:  $\theta_1^c = 0$ ,  $\theta_2^c = \pi$ 
while agents_added <  $n$  do
  Update time  $t \leftarrow t + 1$ 
  Introduce agent  $t$ ; agents_added  $\leftarrow$  agents_added + 1
  Agent  $t$  interacts with agent  $t - 1$  per Algorithm 2
  Update  $n_{t-1}$ ,  $n_t$ ,  $\theta_{t-1}^c$ ,  $\theta_t^c$ , and  $n_{t-1} = n_t = t$ 
end while

```

We derive analytical expressions for $\theta_i^c[n]$; i.e., the centroid locations after all N agents have been introduced in the system.

Lemma 3. *Suppose that $N > 2$ agents are added to the domain $Q = S^1$ as per Algorithm 3. Then, after all N agents have entered Q and pending any further interactions between agents, the positions of the N agents are given by*

$$\theta_1^c[N] = 0, \theta_2^c[t] = \frac{2\pi}{3}$$

$$\theta_p^c[N] = \pi \oplus \sum_{m=3}^p \frac{\pi}{m} \ominus \frac{\pi}{p+1}, \quad p \in [3, N-1] \quad (7)$$

$$\theta_n^c[N] = \pi \oplus \sum_{m=3}^N \frac{\pi}{m} \quad (8)$$

Proof. We prove this result by considering a process wherein, at each time instant $t > 0$, an agent is added to the domain $Q = S^1$. Clearly, at time $t = j$, the j^{th} agent gets added and it communicates only with agent $j - 1$. The domain is partitioned as per Algorithm 2 and the agents move to their individual centroids. This process continues until all agents N are added to the system (i.e., until $t = N$).

At $t = 2$, when agent 2 enters Q , agents 1 and 2 move to locations which are diametrically apart; without loss of generality, we write $\theta_1^c[2] = 0$ and $\theta_2^c[2] = \pi$. Their mutual areas of coverage do not overlap.

At time $t = 3$, agent 3 enters Q and communicates with agent 2; agent 1 does not move. Agents 2 and 3 assign themselves domains of size $2\pi/3$ each; the common boundary of this domain is at the location of agent 2's centroid at time $t = 1$. The new centroid locations are thus given by

$$\theta_2^c[3] = \pi - \frac{\pi}{3} = \frac{2\pi}{3}, \quad \theta_3^c[3] = \pi + \frac{\pi}{3} = \frac{4\pi}{3}$$

This process can be repeated to yield, by induction, that the centroid locations after adding the j^{th} agent at time $3 \leq j < n$ are given by

$$\theta_k^c[j] = \pi \oplus \sum_{m=3}^k \frac{\pi}{m} \ominus \frac{\pi}{k+1}, \quad 3 \leq k \leq j-1$$

$$\theta_k^c[j] = \pi \oplus \sum_{m=3}^j \frac{\pi}{m}, \quad k = j \quad (9)$$

Setting $j = N$ completes the proof. □

3.3.2 Sequential interaction between the agents in Q

Once all N agents have been introduced as per Algorithm 3, we define a sequence of events labeled by $\mathcal{T} = \{k\}$, $k \in \mathbb{N}$ and $k \leq N - 2$, such that at instant k , the agent $N - k$ interacts with agent $N - k - 1$ if $S_{N-k, N-k-1} > 0$ or if $\theta_{N-k-1}^u = \theta_{N-k}^l$ or $\theta_{N-k-1}^l = \theta_{N-k}^u$ (i.e., the borders overlap at one end, which we denote compactly as $\theta_{N-k-1}^{u,l} = \theta_{N-k}^{l,u}$). Else, the sequence is halted. This sequence is enumerated formally in Algorithm 4.

Algorithm 4 Pairwise interaction algorithm - 1

Require: N agents introduced as per Algorithm 3

```

 $k = 1$ ,  $run\_flag = 1$ 
while  $run\_flag = 1$  and  $k \leq N - 2$  do
  if  $S_{N-k, N-k-1} > 0$  or  $\theta_{N-k-1}^{u,l} = \theta_{N-k}^{l,u}$  then
    Update knowledge of  $(N - k - 1)^{th}$  agent
     $\theta_{N-k-1}^c = \theta_{N-k}^c \ominus 2\pi/N$ 
  else
     $run\_flag = 0$ ; halt sequence
    Check for coverage
  end if
   $k \leftarrow k + 1$ 
end while

```

Remark 2. We recall that $\sum_m (\pi/m)$ is unbounded on \mathbb{R} .

$$1 + \sum_{m=3}^p \frac{1}{m} \approx \begin{cases} 1.95 & p = 6 \\ 2.09 & p = 7 \\ 3.999 & p = 50 \end{cases}$$

Thus, on S^1 , after Algorithm 3 is implemented, $j > k$ does not imply that $\theta_j > \theta_k$ (notice the use of $>$ rather than \succ). Informally, if the agents could be viewed as connected by a material thread, this thread could circle S^1 multiple times depending on N (at least twice for $N > 50$).

Remark 2 portends significant complication of our calculations for large values of N . However, it is possible to show that coverage is lost even for a relatively small number of agents. The next theorem shows that the events in Algorithm 4 do *not* lead to an instantaneous loss of coverage if the number of agents is less than 7. It also shows that coverage can be lost for $N = 7$ agents. We generalise this result later.

Remark 3. If the number of agents is capped at $N \leq 7$, it is easy to show that $\theta_j^c[N] > \theta_k^c[N]$ if $j > k$ for $j, k \leq N - 1$. Thus, as long as θ_{N-1}^c does not shift, we can use the usual algebraic operators $+$ and $-$ in place of \oplus and \ominus for calculating the centroid locations resulting from Algorithm 4.

Theorem 1. Let the number of agents be bounded by $N \leq 7$. Suppose that the agents are introduced following Algorithm 3 and they then interact as per Algorithm 4. Then, Algorithm 4 terminates with loss of instantaneous coverage if and only if $N = 7$. Moreover, for $N \leq 6$, Algorithm 4 terminates with an equipartition of Q .

Proof. The case for $N = 1$ and $N = 2$ is trivial and we omit it for brevity. For $N = 3$, at the end of Algorithm 3, $\theta_{\{1,2,3\}}^c = \{0, 2\pi/3, 4\pi/3\}$; $n_{\{1,2,3\}} = \{2, 3, 3\}$. Clearly, $C_2^l = \pi/2 + \pi/3 - 2\pi/3 = \pi/6$ and $\theta_2^l = \pi/3$. When agents 2 and 1 interact based on Algorithm 4, we get that $\theta_1^c = 0$ and $S_1 = 2\pi/3$. Clearly, the domain Q is equipartitioned.

For the case $4 \leq N \leq 7$, if Algorithm 4 terminates only after agents 2 and 1 have interacted, then it follows that all agents have learned about each other; their areas of responsibility thus satisfy $S_i = 2\pi/N$

for all i and that their areas of responsibility do not overlap. Thus, if Algorithm 4 does not terminate before agents 2 and 1 have interacted, then it follows that Q is equipartitioned.

At the end of Algorithm 3, the agents N and $N - 1$ only share a boundary and their partitions are minimally (lazily) sized at $2\pi/N$. Also, let $\alpha = \theta_{N-1}$; this will be a useful anchor for the subsequent calculations, as explained in Remark 3. At time $k = 1$, the agent $N - 1$ interacts with agent $N - 2$; this results in the agent $N - 2$ learning about all N agents and assigning itself an arc of responsibility of size $2\pi/N$. Thus,

$$\theta_{N-2}^c \leftarrow \alpha - \frac{2\pi}{N}$$

The areas of responsibility of agents $N - 1$ and $N - 2$ only share a boundary located at $\alpha - \pi/N$. Suppose that this process continues until time step $p \in \mathcal{T}$, $p \leq N - 3$. Then, at the end of time step p , we get

$$\theta_{N-p-1}^c \leftarrow \alpha - \frac{2\pi}{N}p, \quad S_{N-p-1} = \frac{2\pi}{N}$$

At this point, it is worth noting that

$$\theta_{N-p-2}^c = \begin{cases} \pi + \sum_{m=3}^{N-p-2} \frac{\pi}{m} - \frac{\pi}{N-p-1}, & p < N - 3 \\ 0, & p = N - 3 \end{cases}$$

where we have prescribed that $\sum_3^2(\cdot) = 0$ with slight abuse of notation. If coverage is to be lost at this point, the following necessary and sufficient condition follows from Lemma 1:

$$d(\theta_{N-p-1}^c, \theta_{N-p-2}^c) > \frac{\pi}{N} + \frac{\pi}{N-p-1}, \quad p \leq N - 2 \quad (10)$$

It can be shown readily that Eq. (10) is not satisfied for any permissible p (i.e., $p \leq N - 3$) when $N \leq 6$. Thus, at $p = N - 2$, agents 2 and 1 can interact as per Algorithm 4 to yield an equipartition of Q . On the other hand, when $N = 7$, (10) is satisfied for $p = 4$. Thus, there is no overlap between the areas of responsibility of agents 1 and 2. It follows readily that there is an instantaneous loss of coverage in Q when $N = 7$. This completes the proof. \square

The loss of coverage for $N = 7$ is shown in Fig. 1 which shows the status of coverage at each step of Algorithms 3 and 4.

While Algorithm 4 ensures that coverage is not lost for $N \geq 6$, it is possible to find an alternate sequence of events which lead to loss of coverage for $N = 5$. The sequence of events is enumerated in Algorithm 5. As with the previous sequence, the present sequence starts at the end of Algorithm 3. Thereafter, agent N interacts with agent 1. Formally, we define a sequence of events labelled by $\mathcal{T} = \{k\}$, $k \in \mathbb{N}$ and $k \leq N - 2$, such that at instant 1, the agent N interacts with agent 1 if $S_{N,1} > 0$; and for all subsequent k , agent $k - 1$ interacts with agent k if $S_{k-1,k} > 0$ or if $\theta_{k-1}^{u,l} = \theta_k^{l,u}$ (i.e., the borders overlap). Else, the sequence is halted. This sequence is enumerated formally in Algorithm 5.

Theorem 2. *Let the number of agents be bounded by $N \leq 5$. Suppose that the agents are introduced following Algorithm 3 and they then interact as per Algorithm 5. Then, Algorithm 5 terminates with loss of instantaneous coverage if and only if $N = 5$. Moreover, for $N \leq 4$, Algorithm 4 terminates with an equipartition of Q .*

Proof. The result follows trivially for $N = 1$ and $N = 2$ (where Algorithm 5 is not necessary). The proof for the case $N = 3$ is similar to that for the corresponding case in Thm 1, except that agents 1 and 3 interact instead of 1 and 2. Thus, we need to address only cases $N = 4$ and $N = 5$. At the end of Algorithm 3, we note that

$$\theta_N^c = \pi + \sum_{m=3}^N \frac{\pi}{m} = \begin{cases} 83\pi/60 & N = 4 \\ 107\pi/60 & N = 5 \end{cases}$$

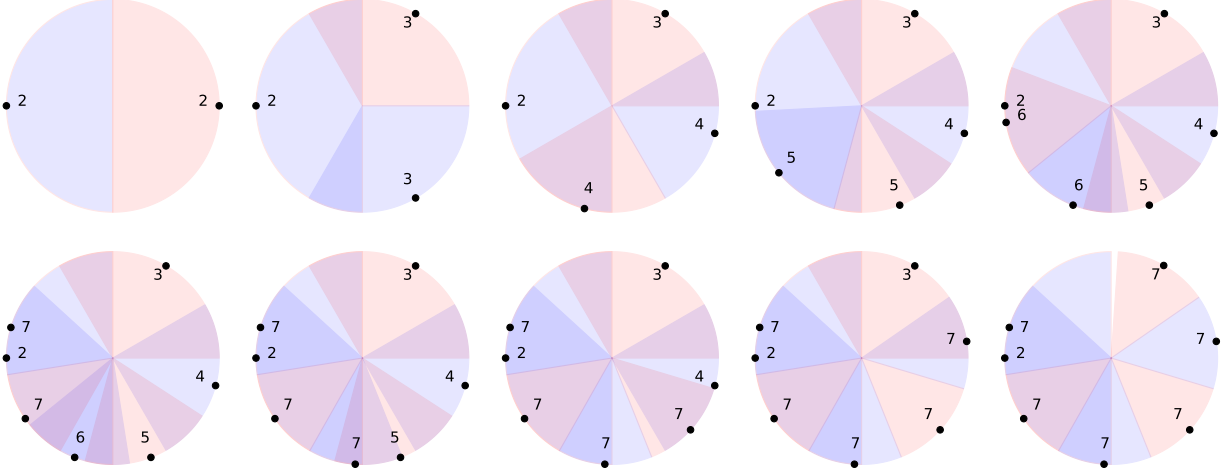


Figure 1: Simulation of 7 agents added sequentially to S^1 . The first 6 images show the addition of agent 2 - 7. Images 7 - 10 show how repartitioning occurs in an anticlockwise fashion, and where failure occurs between two agents (the white region in the last plot of the second row).

Algorithm 5 Pairwise interaction algorithm - 2

Require: N agents introduced as per Algorithm 3

$k = 1, run_flag = 1$

Prescribe agent $k - 1$ for $k = 1$ is agent N

while $run_flag = 1$ and $k \leq N - 2$ **do**

if $S_{k-1,k} > 0$ or $\theta_k^{l,u} = \theta_{k-1}^{u,l}$ **then**

 Update knowledge of k^{th} agent

$\theta_k^c = \theta_{k-1}^c \oplus 2\pi/N$

else

$run_flag = 0$; halt sequence

 Check for coverage

end if

$k \leftarrow k + 1$

end while

for $N \leq 5$. Moreover, the overlap between agents N and 1 ($N \in \{4, 5\}$) can be readily shown to be positive. Thus, at time $k = 1$ in \mathcal{T} , agent 1 moves to

$$\theta_1^c \leftarrow \alpha = \theta_N^c \oplus \frac{2\pi}{N} = \theta_c^N + \frac{2\pi}{N} - 2\pi \quad (11)$$

We have introduced α , as in the proof of Thm 1, to serve as an anchor. In the subsequent interaction p , the agent p interacts with agent $p - 1$ assuming that Algorithm 5 has not terminated prematurely before that

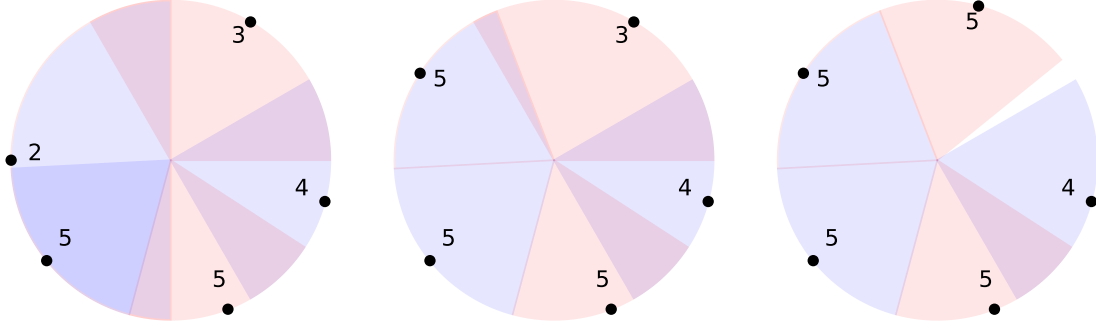


Figure 2: Simulation of 5 agents repartitioning following Algorithm 5. After 2 interactions, coverage is lost between the 2nd and 3rd agent. The size of the uncovered region 9° or 0.157 rad.

instant. If the algorithm first terminates prematurely at p , then a necessary and sufficient condition is that

$$\begin{aligned}
& \frac{\pi}{N} + \frac{\pi}{p+1} < \theta_p^c - \theta_{p-1}^c \\
\Leftrightarrow & \frac{\pi}{N} + \frac{\pi}{p+1} < \pi + \sum_{m=3}^p \frac{\pi}{m} - \frac{\pi}{p+1} - \theta_N^c - \frac{2\pi(p-1)}{N} + 2\pi \\
\Leftrightarrow & \frac{2\pi p}{N} + \frac{2\pi}{p+1} + \sum_{m=p+1}^N \frac{\pi}{m} < 2\pi + \frac{\pi}{N} \tag{12}
\end{aligned}$$

together with $p < N$. It can be checked readily that the condition is not satisfied for $N = 4$ and Algorithm 5 terminates with Q being equipartitioned. For $N = 5$, this condition is satisfied for $p = 3$; i.e., there is an instantaneous loss of coverage between agents 2 and 3. This completes the proof. \square

4 Generalization Using Numerical Experiments

The results presented in the previous section show how a pathological series of interactions can lead to a loss of coverage when the number of agents is small, but still larger than a critical threshold. The same machinery can be extended to cases where the number of agents is larger, but closed-form solutions are not easy to calculate because of the geometric setting of the problem. However, the necessary and sufficient conditions in Eqs. (10) and (12) can be examined through a numerical parametric study for larger values of N than those considered in the previous section. We restrict this study to Algorithm 4 and note that the analysis can be repeated readily for Algorithm 5.

It can be checked readily that the separation between two neighboring agents k and $k+1$ ($2 < k < N-1$) at the end of Algorithm 3 is given by

$$\theta_{k+1}^c - \theta_k^c = \frac{(k+3)\pi}{(k+1)(k+2)}, \quad \theta_N^c - \theta_{N-1}^c = \frac{2\pi}{N}$$

Notice that, for large k , $\theta_{k+1}^c - \theta_k^c \approx \pi/(k+1)$. When N become large, there exists p such that the application of Algorithm 4 and the accompanying interaction between agents $p+1$ and p causes $\theta_p^c < \theta_{p-1}^c$ (informally, agent p “crosses” $p-1$). We refer to this as the C-crossover. There also exists q such that $\theta_q^u < \theta_{q-1}^l$ and $S_{q,q-1} = 0$. We refer to this as the UL-crossover. The crossover index (the agent which crosses over its predecessor) is shown in Fig. 3, with a crossover value of 0 indicating no crossovers. Note that neither of these crossovers corresponds to loss of coverage; the UL-crossover means, in particular, that Algorithm 4 cannot

be applied in its present form once the crossover happens. Moreover, the applicability of the analytical machinery developed in the proof of Thm 1 is restricted to $N \leq 19$.

The application of Algorithm 5 numerically for $N \in [8, 19]$ (the case $N \leq 7$ is covered using Thm 1) shows that Algorithm 5 terminates prematurely with loss of continuous coverage as follows: between agents 1 and 2 for $N \in [7, 11]$; between agents 2 and 3 for $N \in [12, 16]$; and between agents 3 and 4 for $N \in [17, 19]$.

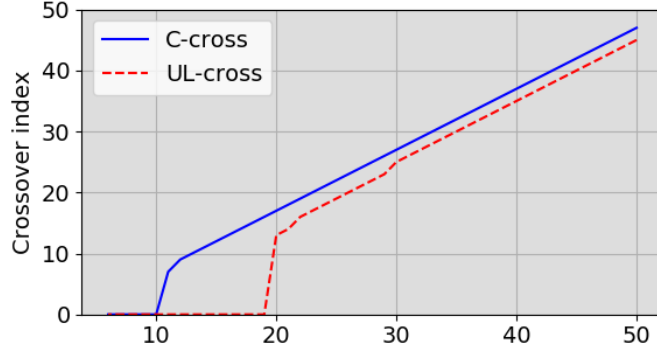


Figure 3: The index of the first agent (starting with N) to cross its predecessor under Algorithm 4. An index of 0 implies that no crossover takes place.

Note that a UL-crossover cannot be avoided when events occur as per Algorithm 4 for large N . We investigate a naive extension whose pseudocode is presented in Algorithm 6. Notice that Algorithm 6 involves a reversion to interaction with the nearest counter-clockwise neighbor and it halts when there is no overlap between an agent and its nearest counter-clockwise neighbor. From Fig. 4 it is evident that the uncovered area reduces with increasing N , although the trend is not monotonic. Although the size of the uncovered area reduces rapidly with increasing N , the partition at the end of Algorithm 6 is seen to not be an equipartition.

Algorithm 6 Naive extension of Algorithm 4

Require: N agents introduced as per Algorithm 3

Require: Algorithm 4 run until premature termination at agent p

```

 $k = 1, run\_flag = 1, p = N$ 
while  $run\_flag = 1$  and  $k \leq k_{max}$  do
  Solve  $p_c = \arg \min_j (d(\theta_p^c, \theta_j^c) \mid \theta_j^c \prec \theta_p^c)$ 
  if  $S_{p,j} > 0$  or  $\theta_p^l = \theta_j^u$  then
    Update:  $\theta_{p_c}^c \leftarrow \theta_p^c \ominus 2\pi/N; p \leftarrow p_c$ 
  else
     $run\_flag = 0$ ; halt sequence
  end if
  Check for coverage
  if Q is equipartitioned then
     $run\_flag = 0$ ; halt sequence
  end if
   $k \leftarrow k + 1$ 
end while

```

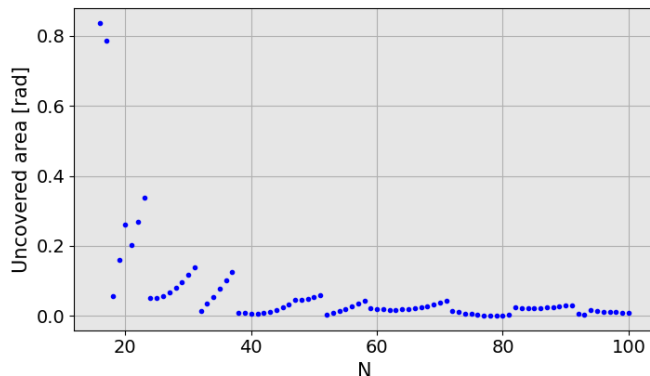


Figure 4: The uncovered area after Algorithm 6 terminates.

5 Concluding Discussion

In this paper, we set out to investigate how the desirable properties of coverage algorithms may change with the number of agents, when the inter-agent communication is discrete, local, and event-driven. We modified a well-known coverage algorithm by prescribing that agents use a certain lazy logic to repartition and resize their areas of responsibility. We applied this algorithm to a simple problem involving lazy agents introduced sequentially into a 2D annular domain. The annular geometry permitted our analysis to be restricted to a unit circle. We constructed a sequence of events that yields an equipartitioned domain for a small but nontrivial number of agents, but fails when the number of agents exceeds a certain threshold. We conducted numerical experiments to demonstrate how the algorithm performs when the number of agents becomes large enough to the point where theoretical analysis is no longer feasible using our methods.

Although carried out in a simplified setting, our work illustrates how the performance guarantees of coverage algorithms can be sensitive to the number of agents, unless the performance guarantees are proven rigorously beforehand for an arbitrary number of agents (which is difficult for general problems involving event-triggered, gossip-based communication). It is not a straight-forward scaling problem, and the actual number of agents plays a critical role in the nature of the guarantees.

We assumed that the agents repartition and resize their areas of responsibility lazily (i.e., with $\epsilon = 0$ in Algorithm 1). Our results show that a degree of altruism might be necessary in order to guarantee coverage using a manifestation of Algorithm 1 that works for an arbitrary number of agents. It remains an open problem to determine if there exists a sequence $\epsilon[t]$ which guarantees that Algorithm 1) leads to an equipartitioned domain for any sufficiently rich sequence of events.

References

- [1] Qier An and Yuan Shen. Distributed coverage control for mobile camera sensor networks with anisotropic perception. *IEEE Sensors*, 21(14):16264–16274, 2021.
- [2] Subhrajit Bhattacharya, Nathan Michael, and Vijay Kumar. Distributed coverage and exploration in unknown non-convex environments. In *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 61–75. Springer, Berlin, 2013.
- [3] Andreas Breitenmoser, Mac Schwager, Jean-Claude Metzger, Roland Siegwart, and Daniela Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *IEEE International Conference on Robotics and Automation, Anchorage, Alaska*, pages 4982–4989, 2010.

- [4] Francesco Bullo, Jorge Cortes, and Sonia Martínez. *Distributed Control of Robotic Networks*. Princeton University Press, 2009.
- [5] Maria Charitidou and Tamás Keviczky. An milp approach for persistent coverage tasks with multiple robots and performance guarantees. *European Journal of Control*, 64:100610, 2022.
- [6] Soon-Jo Chung, Aditya A. Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey of aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.
- [7] Andrew Clark, Basel Alomair, Linda Bushnell, and Radha Poovendran. Submodularity in input node selection for networked linear systems. *IEEE Control Systems Magazine*, 37(6):52–74, 2017.
- [8] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
- [9] Jorge Cortes, Sonia Martínez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(4):691–719, 2005.
- [10] Joseph W. Durham, Ruggero Carli, Paolo Frasca, and Francesco Bullo. Discrete partitioning and coverage control with gossip communication. In *ASME Dynamic Systems and Control Conference, Hollywood, CA*, volume 2, pages 225–232, 2009.
- [11] Anurag Ganguli, Jorge Cortes, and Francesco Bullo. Distributed deployment of asynchronous guards in art galleries. In *Proc. American Control Conference, Minnesota, MN*, pages 1416–1421, 2006.
- [12] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI’07: Proceedings of the 22nd National Conference on Artificial Intelligence*, volume 2, pages 1650–1654, 2007.
- [13] Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pages 2–10, 2006.
- [14] Andreas Krause, Amarjeet Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.
- [15] Benyuan Liu, Olivier Dousse, Phillippe Nain, and Don Towsley. Dynamic coverage of mobile sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(2):301–311, 2013.
- [16] Cameron Nowzari and Jorge Cortes. Self-triggered coordination of robotic networks for optimal deployment. *Automatica*, 48(6):1077–1087, 2012.
- [17] Luciano C. A. Pimenta, Vijay Kumar, Renato C. Mesquita, and Guilherme A. S. Pereira. Sensing and coverage for a network of heterogeneous robots. In *Proc. IEEE Conference on Decision and Control, Cancun, Mexico*, 2008.
- [18] Mac Schwager, Jean-Jacques Slotine, and Daniela Rus. Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3):357–375, 2009.
- [19] Mac Schwager, Jean-Jacques Slotine, and Daniela Rus. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *International Journal of Robotics Research*, 30(3):371–383, 2011.
- [20] Amarjeet Singh, Andreas Krause, Carlos Guestrin, William Kaiser, and Maxim Batalin. Efficient planning of informative paths for multiple robots. In *International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India*, page 2204–2211, 2007.