# Distributed Coverage

**Nov 2023**

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

`project-pavlab/memos/memo_swarm_dist_cover`

## 1 Distributed Coverage

In a class of missions we put under the general umbrella term *distributed coverage*, a core mission goal is to deploy $N$ vehicles to a pre-defined shared given region. A Voronoi partition can be observed from the set of vehicles within this region. Our overall goal is to use the Voronoi partition to inform an algorithm, local to each vehicle, to iteratively adjust its location to a new location that improves certain properties of the partition. The primary property of concern is the area of the local Voronoi cell. Generally, if the variance in areas is small, the distribution of the vehicles over the region is fairly uniform. An algorithm called Lloyd's algorithm can be modified into a vehicle behavior that reasons only about the location of neighbors, thereby forming a belief about its own local cell, without awareness of the entire Voronoi partition. Llyod's algorithm proceeds by iteratively moving each vehicle toward the center of its own Voronoi cell. Variations of Lloyd's algorithm can be customize to achieve different performance metric, or even perpetuation motion of the field, to enhancing sensor performance. The basic idea is shown in Figure 1 below, from our initial development of the algorithm in an engineering GUI (the `voiview` application in the Swarm Autonomy Toolbox).

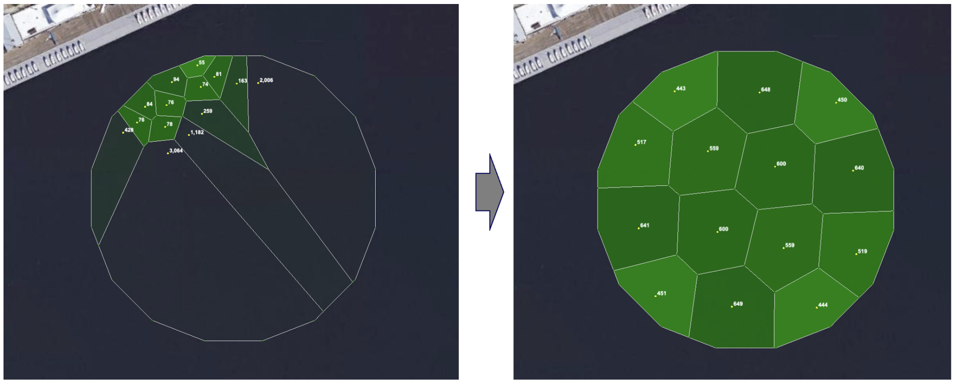

Figure 1: **Lloyd's Algorithm Variation Engineering Model Simulation**: Left: An initial distribution of points in a convex polygon, akin to an initial vehicle deployment in a search region. Right: The distribution of points after several iterations of Lloyd's algorithm.

The algorithm reaches a stable state when all nodes have simultaneously reached their set point (by default the centroid of each node's Voronoi cell). A tolerance range can be configured, to allow the nodes to achieve their stable state sooner.

In distributing the vehicles, the goals, or performance metrics, are to (a) minimize the variance in area of each vehicle's Voronoi cell, (b) minimize the time for the group to achieve a steady state, (c) minimize the distance traveled for each vehicle, and others. There are three mission variations that have provided motivation due to needs from related ongoing efforts. These are discussed next.

## 1.1 Distributed Environmental Monitoring

In the first application area the goal is to deploy multiple vessels to a region, each equipped with environmental sensors. In certain aquaculture installations, real time environmental monitoring is highly valued. We are working with a small sailboat company, Marine Robotics, LLC, in Maryland who engage with oyster farmers in Chesapeake Bay. We are developing the distributed coverage algorithms described here along with autonomous sailing methods. MIT has had a single autonomous sailboat on campus since Fall of 2021, with the goal of deploying multiple vehicles in the next year.

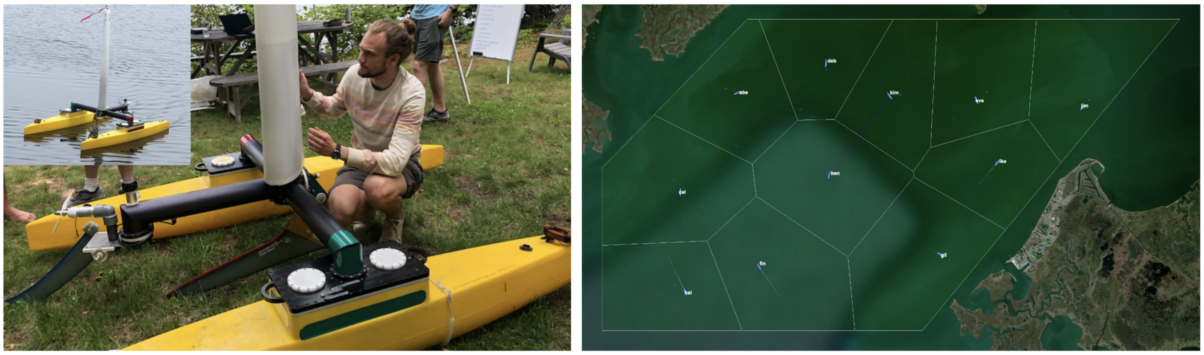

Figure 2: **Distributed Environmental Monitoring**: Left: An autonomous sailboat by Marine Robotics LLC in Maryland. Right: A simulation of 11 vehicles in a region near Saxis Maryland near an oyster farm installation.

In this application the initial objective is to achieve a Voronoi covering of the region of interest, optimizing the metrics of minimizing Voronoi cell variance, travel time, and travel distance. The extra challenge lies in achieving these objectives with a sailing vessel which is limited by the prevailing wind direction and magnitude . The current approach is to implement both components of the algorithm as distinct IvP behaviors, each optimizing their respective performance metrics, reconciled by the multi-objective optimization engine of the IvP helm.

## 1.2 Distributed Sensing for ASW

In the second application area involving the Voronoi family of algorithms, the goal also begins with deploying multiple vessels to a region. Rather than being deployed with environmental sensors, they are deployed with sensors for detecting other vessels moving through the region. This is essentially to replicate a distributed anti-submarine warfare (ASW) domain.

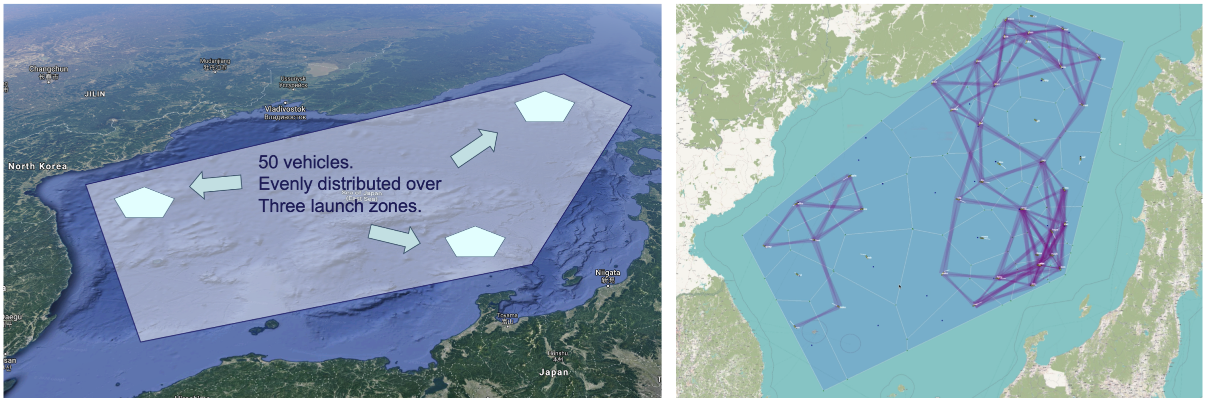

Figure 3: **Distributed ASW Mission**: Left: A simulated mission involving the Sea of Japan. Fifty vehicles are deployed from three distinct launch points. Right: A simulation of the 50 deployed vehicles, at a point of near uniform coverage. The simulation ran on a cluster of 50 networked RasPi computers, periodically sharing position information between neighbors.

One of our US Navy graduate students, LT Craig Evans, is developing missions based on this scenario to include simulated sensors, and randomly placed moving contacts through the field. Automated random simulations are conducted, varying sensor performance, target speed, and different patterns of movement for the deployed vehicles. LT Evans is expecting to complete his work in August 2022.

## 1.3 Multi-Vehicle Mustering

In the third application area involving the Voronoi family of algorithms, the goal does not involve distributed sensing. In this case the goal is to deploy $N$ vehicles to a *muster region*. This region is also a convex polygon as in the previous two mission types, but in this case the region is much smaller, and the goal is to transit the group of vehicles to the region and muster, or wait until a follow-on mission mode. The idea is to use the Voronoi algorithms in a Muster behavior to ensure vehicles maintain a separation from each other rather than relying purely on a collision avoidance behavior. The collision avoidance behavior is still used, but the Muster behavior helps avoid situations where collision avoidance measures are needed to begin with.

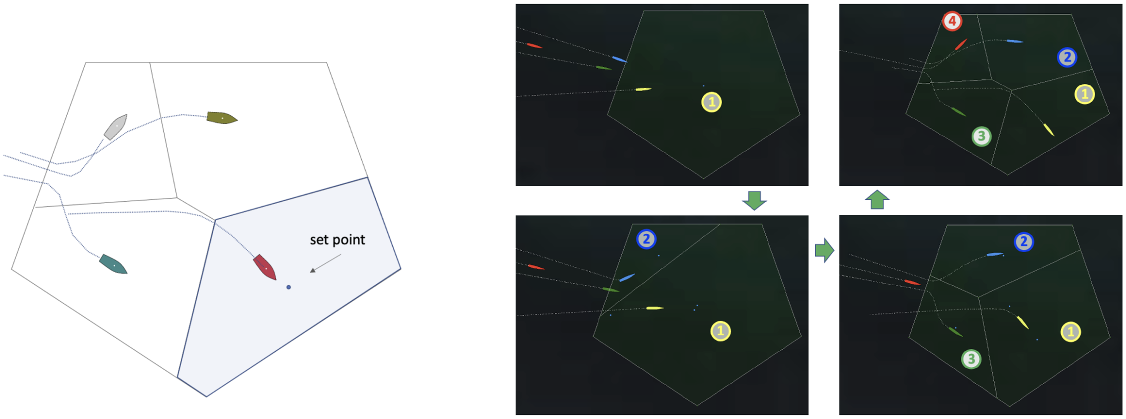

Figure 4: **Four Vehicle Muster**: Four-vehicle Muster Four vehicles approach a muster region. As each vehicle enters the region the Voronoi decomposition is shown. When the first vehicle enters, there is only one region. When the second vehicle enters, there are two regions, and so on. Upon each new vehicle, the local regions for each vehicle may change. The change in local region shape affects the set point toward which the vehicle is driving toward.

Compared to methods that assign a wait point to each vehicle, the muster behavior is preferred for two reasons (a) the single muster region is a simpler mission plan to construct that scales with the number of vehicles, and (b) with the single wait-point method, occasionally a vehicle will need to maneuver around other vehicles that arrived at their wait point earlier.

This behavior is defined in the `lib_bhv_muster` library. It has been field tested on four Heron USVs on the Charles River, and has been integrated into the DARPA Sea Train autonomy system for mustering the four vehicles in several mission sub-modes for that project.