DronePollinationProject

Abolfazl Razi

April 2023

1 Introduction

In recent years, populations of natural pollinators have declined as demands on agriculture continue to rise.

2 System Model

We assume that the wind created by the blades of a drone has an effect radius, and that any plant within that radius is being shaken. We also assume that each plant must be shaken for a certain amount of time is considered fully pollinated. That is, that shaking it any more with the drone would be no more effective at spreading pollen to its neighbors.

The field over which the drones is pollinating is modeled simply as an array of points. These points are generated with the Scikit-learn make_blobs method. This way, there are more and less dense areas of flowers, and the distribution looks somewhat natural. The field models are also square, but their dimensions can be specified by the user. An example of what a distribution of the flowers look like can be seen in the following figure.

The drone swarm used to pollinate this field is also modeled as a series of points. Each drone is given a position and a velocity vector. Each drone has two radii. One is for the area of flowers which it can shake with its blades, denoted R_e. One is for the the area of flowers which the drone can see in its camera, denoted R_c. Each flower that is seen by a drone has its position stored in a dictionary. This dictionary maps the flower's position to the amount of time it has been shaken by a drone. This dictionary is shared across the swarm, and is not unique to each drone.

2.1 Mean-Shift Algorithm

As such, each drone's movement is informed by the mean-shift sliding window algorithm. Flowers that have been shaken for less total time can be prioritized less than those that have been shaken for longer. As such, each drone's velocity is calculated as follows.

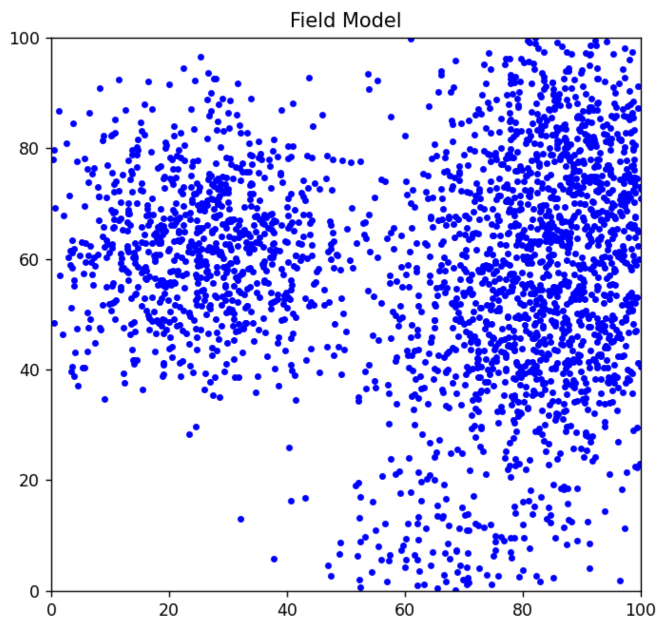

Figure 1: An example of a model of a field of flowers.

$$\vec{V_j} = \frac{\sum_{R_c} w_i * \vec{r_i}}{\sum_{R_c} w_i} \tag{1}$$

Here, the velocity vector of the jth drone is denoted as $\vec{V_j}$. The positions and weights of all flowers within the radius R_c of the position of the drone, $\vec{r_i}$, are considered. The positions of the jth flower is denoted as $\vec{r_j}$, and the weight of the jth flower is denoted as w_i. Each flower's weight is calculated as follows.

$$w_i = max(t_{fs} - t_j, 0) \tag{2}$$

Here, the t_{fs} denotes the time for the flower to be considered fully shaken, whereas t_j denotes the time that the jth flower has been shaken. In this way, flowers that have been shaken for a time of t_{fs} or longer have a weight of 0, and aren't considered at all in the calculation of the drone's velocity.

Under this algorithm, when $\sum_{R_c} w_i = 0$, or when the drone sees no flowers with any weight in its camera, it will continue with a constant velocity until it finds a flower with some weight, or until it reaches the edge of the area designated as the field. If a drone reaches the edge of the area designated as the field, it moves away from that edge at the reflection angle of the velocity it had when coming to that edge.

On its own, this methodology of determining drone's velocities has them moving until they find a dense area of flowers, and then being still until that area is considered fully shaken. Once several drones have discovered a cluster of flowers, they do well in exploring and pollinating all of the flowers in that cluster. However, if one watches the drones through the animation functionality of the simulation, one will notice that the drones spend a large portion of time wandering around areas which either have no flowers in them, or in which the flowers are fully pollinated. For this reason, Particle Swarm Optimization was introduced to solve this problem.

2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) was introduced as follows. Each drone has a bias in its velocity towards a weighted mean position of all the drones. This is calculated as follows.

$$\vec{R_{PSO}} = \frac{\sum_i \vec{R_i} w_i}{\sum_i w_i} \tag{3}$$

$$\vec{V_{PSO}} = \vec{R_{PSO}} - \vec{R_i} \tag{4}$$

3 Citations

https://www.beeculture.com/synthetic-pollination-invasive-species/