Evolving Optimal Parameters for Swarm Control

Jennifer Golbeck
University of Maryland, College Park
A.V. Williams Building
College Park, Maryland 20742
golbeck@cs.umd.edu

Abstract

Using many inexpensive rovers in place of single costly ones is an idea that has been gaining attention in the last decade. How to effectively control this hardware is an open question, but because of its efficiency and distributed nature, swarming is an attractive option. While much research in the field investigates intelligent swarming, recent research has shown that the “unintelligent” swarm is an effective control mechanism for thoroughly covering a space and maintaining swarm-like behavior in the face of widespread failures. This paper takes that research one step further, exploring the application of a genetic algorithm to evolve optimal parameters for an exploratory swarm.

1. Introduction

When rovers need to cover ground, it is important to think about how to organize and control them. Centralized control can be expensive and difficult. Furthermore, if the central control system fails, the entire network of rovers is incapacitated. Using a deterministic plan for covering ground is also susceptible to faults – if a rover meets an obstacle or if it is damaged, the unfinished portion of its route may not be covered. To the contrary, an approach that mimics the swarming seen in nature avoids all of these pitfalls. The built-in redundancy and non-reliance on any individual makes a system that is highly fault tolerant and balanced between exploration and localization.

Previous research has investigated the unintelligent swarm as a control mechanism. That research showed that the swarm was naturally an effective strategy for covering territory with a normal distribution as well as dealing with large failure rates within the system [6]. An unanswered question is what makes an ideal swarm? What is the best way to organize a swarm internally? This research employs a genetic algorithm to evolve optimal parameters for a swarm of rovers engaged in an exploratory mission.

2. Experiment and Results

Each individual in the swarm here moves independently. The position to which an individual moves is the composite vector of the nearest neighbors and a vector of acceleration toward the middle.

A standard genetic algorithm with elitism was used to optimize five parameters of the swarm.

1. Number of neighbors – ranging from 2, the minimum required to achieve swarming behavior, to $n$ where $n$ is the number of rovers in the simulation.
2. Acceleration to center – how strongly each rover is drawn back toward the center of the area.
3. Attraction toward neighbors – how strongly each rover is pulled in the direction of the nearest-neighbors vector.
4. Randomization frequency – how often neighbors are randomized to prevent clustering.
5. Repelling distance – how close rovers can come to one another before they start to repel. This metric prevents collisions, and also prevents a number of rovers from investigating a small area at the same time.

Repeated simulations were used to analyze the effectiveness of the genetic algorithm. To start, simulations used ten rovers. A population of parameter-sets was randomly initialized. The area to be explored was broken into 20,000 X 40,000 units. Ten targets, each measuring one square unit, were randomly placed throughout the space. The fitness of a population was determined by how many targets were discovered and how often each was visited. The GA proceeded with roulette wheel selection with elitism and midpoint crossover reproduction. A 1% mutation rate was introduced to the reproduction phase.

With the initial conditions described above and a population of 20 parameter-sets, there is a significant increase in fitness over 25 generations of evolution. Figure 1 shows the average fitness at each generation.