Angry Ants: Progress Report

Jasmin Uribe and Yunhao Xu

April 24, 2013

1 Jan 29 2013

- Updated project plan document, including locating (a subset of) relevant papers
- Obtained data file from Yunhao, started writing code to process data
- Started preliminary work on pilot experiments for the MC Basic algorithm
- Began thinking about experiments that need to be run in order to implement shotgun algorithm

2 Feb 5 2013

- Continued writing code to process data
- Discussed what information the trajectory graph needs to contain and how to obtain this from the data files
- Discussed Alon's algorithm tried generalize to an arbitrary number of ants
- Began formalizing shotgun sequencing algorithm, still brainstorming on a good way to initialize ants at different start times

3 Feb 12 2013

- Continued writing code to process data; code takes in trajectories in text files and finds clusters (ie. intersection points)
- Currently working on processing these clusters into a graph
- Found python library for data analysis including min-cost
- Discussed clustering parameters, ie. how close is considered an intersection point

4 Feb 19 2013

- Continued working on processing path clusters into the trajectory graph
- Explored python library for data analysis
- Discussed how to test min-cost and greedy algorithms (current data set is essentialy only 1 'experiment', need more data sets)

5 Feb 26 2013

- Continued working on creating trajectory graph
- Fleshed out shot-gun sequencing method (see below)
- Got access to local methods, will begin running and exploring the code

5.1 Shot-gun sequencing method

Data must be gathered differently in order to implement this method. We already have 'ground zero' trajectories for a second data set, we will use this to initialize each frame for the shot-gun method. We will randomly pick an ant to track (discrete uniform distribution). Using the ground truth trajectory for this ant, we will randomly pick a time/location to initialize the game (discrete uniform distribution). The user will then track the ant from the specified time/place for a specified lenth of time. We will begin by having all users track for a fixed length of time (ie 30 sec), then we will have users track for a randomly distributed length of time (Gaussian distribution). For each ant we will continue initializing at a new time/place until each time step has been tracked at least 3 times (note: this parameter may change depending on data). Data for each ant will consist of a number, n, of mini-paths. Using these mini-paths and the time correspondence between them, we will 'glue' the mini-paths together creating the final trajectory of the ant. We will determine the accuracy of the method by examining the Frechet distance and the least squares distance from the ground truth.

5.2 Other Methods

Along with the shot-gun method we will also find the optimal trajectory for each ant using a simple greedy algorithm, a min-cost algorithm and the Sergei/Alon linear programming algorithm. We will compare the results of all these algorithms, as well as the results from the local algorithms, to the ground truth.

5.3 Short term goals

- Explore local algorithms
- Email/meet with students working on Ants Game-provide them with specific details on how to initialize game for shot-gun method
- Finish creating trajectory graph for Data set 1 and run simple greedy and min-cost algorithms

5.4 Longer goals

- Figure out exact details and write algorithm for how to process shot-gun data into full trajectories
- Generate random data set to test shot-gun method
- Run Sergei/Alon linear programming algorithm on data set 1, compare local/global for data set 1
- Once data set 2 collected, run all algorithms on data set 2 and compare

6 March 19 2013

- Read a multitude of papers relevant to research goals
- Finished creating trajectory graph algorithm
- Explored local methods
- Explored implementing algorithms in Wiratma paper to use as comparison
- Figured out what criteria defines a 'good' median and how to measure
- Started measuring this criteria in implemented methods

7 March 26 2013

- Continued work on generating data similar to 'real' data in order to test global algorithms
- Set up meeting with Sergey to discuss data generation
- Began searching for papers that include temporal component

8 April 2 2013

- Met with Sergey, Sankar, Lukas to discuss generting data-look for work on simulating paths
- Looked at algorithms that generate 3D random walks–maybe use this to generate data
- Met with Yunhao and discussed details about the project
- Looked at current algorithms and discussed how portions can be used for our project
- Set up goals for the week

9 April 9 2013

- Work on the new ant data, create underline graph for greedy and ILP algorithm.
- Implement the greedy algorithm, prepare to compute the trajectories. (Have some bugs, not quite finished)
- Implement underline graph generator, including the ground graph and the user points.
- Compare underline graphs created by clustering algorithm with different threshold.

10 April 16 2013

- Finished implementing greedy algorithm
- Created semi-random graph generator that forces an intersection at each time step
- Created global graph using new data set- checking for possible bugs
- Plan of attack- create random graphs and semi-random graphs, create user selections, use local algs, greedy and fuzzy k-means to extract median paths and compare to ground truth

11 April 23 2013

- We analyze our output from cluster & greedy approach on the real data. According to the result, we believe this approach is at least as good as the automated and local approaches. (See Figure 11)
- Generating plots, prepare for the final report.
- We get around the "off screen" problem, by manually find out these frames and remove them away for particular ants. So if an ant went off screen between frame i to j, we make frame j+1 be the next frame of i-1 for that particular ant. We plan to get the data from the newest game with "off screen" option implemented, and run our test on that.