DRONE COLLISION DETECTION SYSTEM

ABSTRACT

Help the drone to travel to the destination with efficient path.

Snehal Bhilare

INFO6205-35206 Program Structure & Algorithms

Under Guidance of

Prof. Kal Bugrara

Problem Statement:

There is flying drone and we want to monitor their progress toward their target destination. Required to design algorithms to detect anomaly in their flight path and issue corrections so the drone can arrive its target location. Also machine learning techniques are needed to solve this problem.

Predicated Solution:

1) We will be storing the paths which are successfully travelled in the past.
2) This machine will be used to simulate drone movement.
3) With given source and destination, best possible path is detected from machine and drone is set for the found path. If there is no historical data available drone will move to the node which is having minimum distance from the destination.
4) We will also consider anomaly constrain before moving to any node. There can be two types of anomaly with high and low risk.
5) If high intensity anomaly is present at the next node then drone is not permitted to travel same. Rather strategy is refined again to find next possible node. For the low intensity drone is set to move to that node but fuel is reduced by *3.
6) After reaching to destination, all new paths discovered by the drone will be added to the machine to improve simulation techniques.

Techniques Used:

Data Structure: Hashtable and Priority Linked List

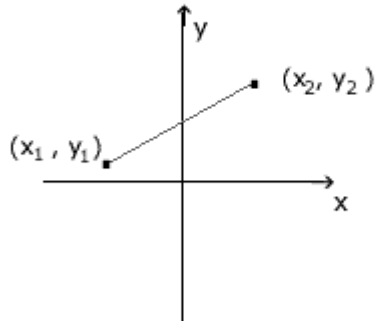
All the traversed nodes are stored into a path. Each source can have multiple paths by which drone can reach to the destination. These paths are stored into priority linked list with maximum travelled path (giving safest solution) as a root. Hashtable is used to store all the possible combination of sources and its paths. Priority linked list is inserted against each source which acts as a key to hashtable. Hence hashtable uses **separate chaining** method.

Combination of priority linked list and hashtable provides **O(1) searching complexity**.

Algorithm: Nearest Neighbour Algorithm:

The algorithm starts at a random node and repeatedly visits the nearest node giving optimal distance from the destination. Distance is calculated by below formula.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



The nearest neighbour algorithm is easy to implement and executes quickly.

Machine Learning Technique: Weighted Majority Algorithm and Ada Boost

Weighted Majority Algorithm

The algorithm assumes that we have no prior knowledge about the accuracy of the algorithms in the pool, but there are sufficient reasons to believe that one or more will perform well. To train machine a positive weight is given to each of the travelled path. The compound algorithm then collects weighted priority from all the paths in the pool, and gives the prediction that has a higher priority.

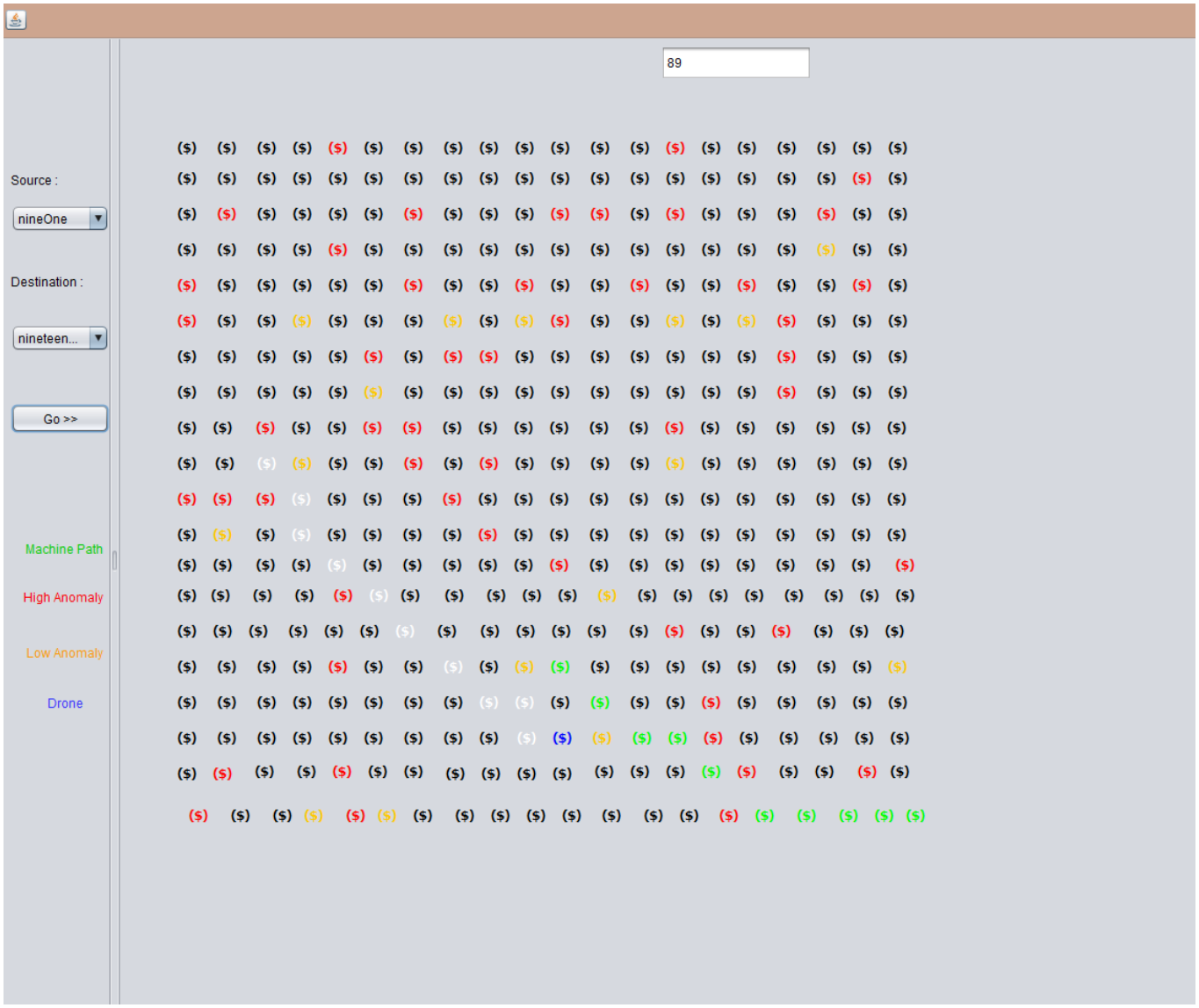
Ada Boost

AdaBoost, short for "Adaptive Boosting", is a machine learning meta-algorithm. It gives a weighted sum that represents the final output of the boosted classifier. All the nodes are adjusted before achieving optimal performance on a dataset using priority. When used with decision learning, information gathered at each stage of the program and fed into the machine.

Collision Handling:

We have considered two types of anomalies. High Intensity and Low Intensity. Every second anomalies are generated randomly. Drone cannot move to the node which has high intensity. This collision is handled using nearest neighbour algorithm. Whenever we are facing high intensity anomaly, the adjust node with least distance from the destination is calculated and drone is deviated to that node.

Screenshot:



References:

- https://www.youtube.com/watch?v=5mBiac_dhbs
- http://en.wikipedia.org/wiki/Nearest_neighbour_algorithm
- http://en.wikipedia.org/wiki/AdaBoost
- https://firstlook.org/theintercept/2014/02/10/the-nsas-secret-role/
- http://en.wikipedia.org/wiki/Hierarchical_temporal_memory
- http://math.about.com/library/bldistance.htm
- http://en.wikipedia.org/wiki/Weighted_Majority_Algorithm