**Initial Distance Minimizer Script**

**Structure**

* Randomly choses an action to move NESW
* When action brings it closer to target, increases likelihood of that action in the future, otherwise decreases
* Run within SITL simulation

**Basic Action Weight Shifting**

* Start equal (weight list is (25, 25, 25, 25))
* If moving north brings you closer to the target, shift a constant weight from S to N
* Works sometimes for constant weight shift of 10
  + Tests run on returning home after moving 20 m N, 25 m E
  + Gets back to home in a couple minutes ideally
  + Can often overshoot target, BIG PROBLEM
  + Note: Weight list doesn’t have to add to 100, they are just relative weights. Weights can be negative.
* Countering Overshoot issue
  + Cap weights
  + Change action distance according to proximity

Basic Action Weight Shifting with Capped Weights

* 1 for 1 successful returns to home with weight range (10, 40)
  + 340 seconds
  + Effective range is (15, 35) due to starting weights of 25 and jump size of 10
* 3 for 3 successful returns to home with weight range (5, 45)
  + 280, 277, 262 seconds
  + Correctly handled overshoot!
* Slower than ideal non-capped run because there is always a chance of backtracking

Basic Action Weight Shifting with Capped Weights and Proximity-Based Action Distance

Scaled Action Weight Shifting

* Weight shift scaled based on some impact factor
* Impact factor can be something like change in distance