Jeffrey Hatton

Parallel Computing

9/14/15

1. For my enhancements ant-sim was to add two different types of ants, hauler and seeker ants. The seeker ant’s primary responsibility is to find for the nest and the hauler ants primary reasonability is to bring the foods back to the nest. The seeker ants will leave the nest first and will tend to go to cells that have the least pheromones in front of them. Once the seeker ant finds food it will return to the nest releasing much more pheromones along the way. Hauler ants won’t leave the nest until food has been found. Once food has been found hauler ants will leave the nest and tend to go to cells with the most pheromones. Once the hauler ant finds food it will return to the nest releasing more pheromones. Both ants repeat their behavior indefinitely.   
    The main idea behind these two ants is that the seeker ants would always be in search of unexplored areas or areas that were explored long ago. Once they found food they would return to the nest and leave a strong trail for the hauler ants to follow. The hauler ants would then reinforce this trail by collecting food from the strong trail while other less traveled trails would dissipate and become less likely to be traveled. This would allow quick collection of a single food source. I made it so that there would be exponentially more hauler ants than seeker ants so the retrieval of food would keep up with the finding of food. The ants on the outside edge of the square are seeker ants while the rest are hauler ants. Seeker ants are blue while hauler ants are green.
2. Algorithms
   1. Seeker Ants
      1. Check if the ant has food
         1. If it does return to the nest release X more pheromones
         2. If not check the cells in front of the ant for the least average pheromones
            1. Average pheromones is calculated by finding the average pheromone count in an X by Y rectangle including the cell being examined. Check the end of the document for a diagram.
         3. The cell the ant goes to is then randomly determined with more weight given to the cells with the least pheromones.
         4. Drop pheromones in the new cell
   2. Hauler ants
      1. Check if food has been returned to the nest
         1. If food has not been returned do nothing
         2. If food has been returned check if the ant has any food
            1. If the ant has food return to the next releasing x more pheromones
            2. If no food randomly choose a cell in front of the ant with more weight given to cells with more pheromones.
   3. Overhead
      1. The enhancements do not add much overhead to the original algorithm. The hauler ant only adds one more check compared to the original algorithm. The seeker ant add the majority of the overhead with its 3(X\*Y) sums when it tries to find the least average pheromones. Neither the seeker nor the hauler ant add any more concurrency overhead over the original algorithm.
      2. Final Overhead
         1. Both
            1. One check for type of ant
            2. Parallel currency

Possible memory write fails when moving ants

* + - 1. Seeker Ants
         1. 1 check for food
         2. X \* Y sums and one division for the average pheromone check
         3. Overhead for random cell selection
         4. Two writes one for the location of the ant and the pheromones level
      2. Hauler ant
         1. 1 check for if food has been returned
         2. 1 check for if the ant has food
         3. Overhead for random cell selection
         4. Two writes one for the location of the ant and the pheromones level