# Computational Intelligence: Report assignment 3

Tjitte de Jong - 4172930
Boris Mulder - 4100794
Max Spanoghe - 4331834

October 29, 2014

# 1 ANT PREPARATION: OBSERVE THE PROBLEM

Here we will answer the questions about the observation of the problem.

## 1.1

First of all it could be possible that the ant is in a very big room. This would be difficult because it is hard to find the exit and the ant can make an unnecessary turns which can lead to a very long path. Secondly, it could happen that the ant gets in a dead end and needs to go back all the way to a previous choice point.

## 1.2

The ants drop Pheromones to let the other ants know the popularity of a certain path. The amount of pheromone dropped is dependent on how long the path is that he took. The equation which determines the amount of pheromones par path is:
$Pheromone_on_link_i = \frac{n}{Pathlength}$ The parameter n is the estimated pathlength.

## 1.3

The evaporation of a link i is is actually the Pheromones that are getting away from that link. For example:
evaporation constant $\alpha = 0.1$;
Pheromones on link in iteration i = p ;
Then
Pheromones on link in next iteration = (1-$\alpha$)*p;
The evaporation is very handy to get a more quick convergence to a certain path. If a path it not chosen the links on that path get less pheromones after each iteration. In this way, the chances that ants take a path that is not very popular decreases much faster. Each iteration the pheromones go down relatively because they don't take the path, but also because of the evaporation.

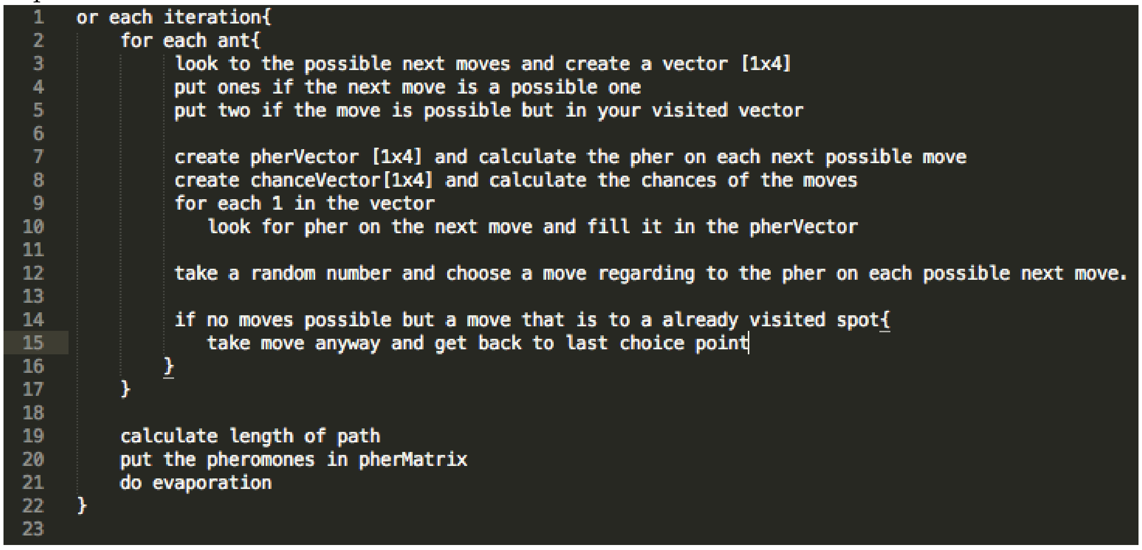
# 2 IMPLEMENTING SWARM INTELLIGENCE

Here we will answer the questions about the actual implementation.

## 2.1

The following parameters are used in the basic version of our system:
$iterations = 1000$;
$ants = 10$;
$pher = 100$;
$evaporation = 0.80$;
converge criterion is when the iterations are done or when the probability of a certain path is higher than 0.99.

pseudo-code:



```
or each iteration{
    for each ant{
         look to the possible next moves and create a vector [1x4]
         put ones if the next move is a possible one
         put two if the move is possible but in your visited vector

         create pherVector [1x4] and calculate the pher on each next possible move
         create chanceVector[1x4] and calculate the chances of the moves
         for each 1 in the vector
            look for pher on the next move and fill it in the pherVector

         take a random number and choose a move regarding to the pher on each possible next move.

         if no moves possible but a move that is to a already visited spot{
            take move anyway and get back to last choice point
        }
    }

    calculate length of path
    put the pheromones in pherMatrix
    do evaporation
}
```

## 3 UPGRADING YOUR ANTS WITH INTELLIGENCE

Here we will answer the questions about how to upgrade the intelligence of the ants.

### 3.1

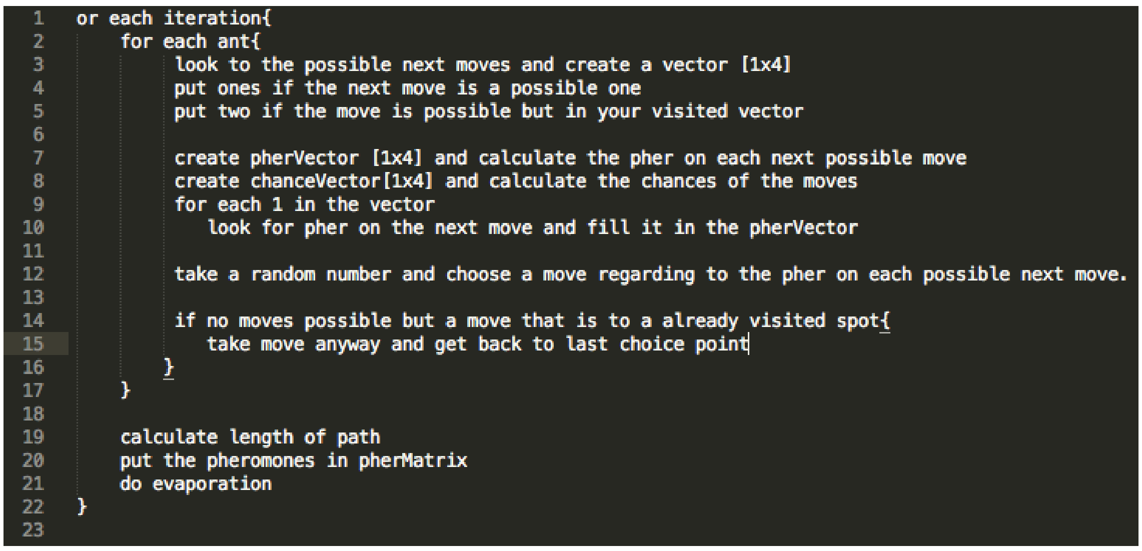
Especially for the big rooms we have found a better solution than the normal maze solving algorithm we had before. The way it works is pretty good and still easy enough to implement quickly. If the ant enters a big room we do a check. If the ant has four possible next moves he will do this as long as he has 4 next possible moves:

take a direction just based on the normal solution and forbid him to take the opposite direction again. In this way, he will move but never have loops in a big room. It stops when he hits a wall, bounce and will do it again eventually. This method prevent the ant of doing stupid loops in a big room. For example, if YOU are in a big room and you cannot see the actual walls. What one would do is start walking until he sees a wall, but never move back until you have found one, since you know there is nothing there but open space. Once you see a wall and you see there is no exit, you will bounce and start walking in the other direction again or just follow the walls. This is actually what this algorithm does so it actually works like a real person would solve the maze big rooms.

## 4 UPGRADING YOUR ANTS WITH INTELLIGENCE

## 5 PARAMETER OPTIMIZATION

## 6 PART 2: THE TRAVELING ROBOT PROBLEM



```
or each iteration{
    for each ant{
        look to the possible next moves and create a vector [1x4]
        put ones if the next move is a possible one
        put two if the move is possible but in your visited vector

        create pherVector [1x4] and calculate the pher on each next possible move
        create chanceVector[1x4] and calculate the chances of the moves
        for each 1 in the vector
           look for pher on the next move and fill it in the pherVector

        take a random number and choose a move regarding to the pher on each possible next move.

        if no moves possible but a move that is to a already visited spot{
           take move anyway and get back to last choice point
        }
    }

    calculate length of path
    put the pheromones in pherMatrix
    do evaporation
}
```

Figure 1: pseudo code for ant colony optimization