Ant Foraging Optimization through Artificial Pheromone System

Aman Timalsina

State University of New York at Buffalo, amantima@buffalo.edu

Abstract

In this paper, the foraging pattern found in ants and other 'swarm intelligent' insects is simulated using random walk model. The first iteration uses the ants' odor memory to influence each ants' choice of moving in either of the four directions. As expected, the steps taken by the colony gets reduced significantly and a pheromone trail is updated based on the grid locations each ant has to traverse and its last location. Latter iterations show that simulations aided by both odor memory and pheromone trails optimize the steps the colony has to take to reach to the food.

Intoduction

Several ant colony optimization algorithms (ACO) exist in computer science and operations research that address the question of how a collective intelligence emerges from self-organizing societies of ants in food foraging. A common factor in all of these algorithms is the pheromone trail. Pheromone is a secreted chemical that is used to communicate between members of the same colonies. Previous foragers leave this chemical for the next round of foragers so that they use specific paths from the nest to the food location. Over time, pheromone trails start to evaporate so that the convergence to a locally optimal solution can be avoided. More importantly, pheromone evaporation helps in establishing shorter paths as these paths develop higher phermone density.
Moreover, an overlooked factor in most of these algorithms is the odor memory of an ant. As the researchers from the Max Planck Institute for Chemical Ecology stated in their 2018 paper, an ant learns to stor food odors in their memory even after a single contact. Therefore, to achieve a closer simulation to the natural ant foraging pattern, the first iteration uses the strengh of smell calculated through a Cartesian distance function from the current location to the food location at every step. Finally, a random walk model is chosen as at first, ants do roam randomly. However, albeit being effective in territorial search, at a certain stage, ants use their memory (odor) and experience (pheromone trails) to

```
 navigate.
This paper seeks to simulate these exploration patterns through several
 iterations using a distance-based odor influence and an artificial pher
omone system.
```

Methodology

```
As mentioned above, the random walk model is a simple 2-d random walk mo
del with each ant having an equal probability of moving in either of the
four directions: Up, Down, Right, or Left. However, at a fixed distance
 from the food location (35), their odor memory starts to get triggered
 by the smell from the food. This influences the "size" of steps taken f
or the "up" and "left" probability. An expression that calculates the lo
garithmic distance from the food to the current location is added to the
se probabilities. That is;
```

$$strength_{smell} = \frac{10^{3.71117}}{\left(\sqrt{(loc_x - 10)^2 + (loc_y - 10)^2} + 1\right)^2}$$

$$location_{Current} = location_{Previous} \pm 1 \pm int(\log(strength_{smell}))$$

```
After the first iteration is concluded, a pheromone trail is updated aft
er the strenth of pheromone in each grid is calculated by the final posi
tion of the ant that took that particular path. Further iterations use t
hese trails in determining the size of steps to be taken while moving up
and left. An evaporation coefficient is also added where after each iter
ation the pheromone is updated by:
```

$$\tau_{xy} = (1 - \rho)\tau_{xy} + \sum_k \Delta\tau_{xy}$$

where τ_{xy} = the amount of pheromone deposited for a state transition,

ρ = the pheromone evaporation coefficient,and

$\Delta\tau_{xy}^k$ = the amount of pheromone deposited by kth ant.

```
Moreover, each iteration's size of step for "up" and "left" direction is
modified as:
```

$$strength_{pheromone} = Array[loc_x[\min(\max(i, 0), 99)], loc_y[\min(\max(i, 0), 99)]]$$

$$strength_{overall} = strength_{pheromone} + strength_{smell}$$

$$location_{Current} = location_{Previous} \pm 1 \pm int(\log(strength_{overall}))$$

```
Finally, the simulation stops if any one of the ants gets at least 5 uni
ts close to the food location.  A pheromone update is carried before the
code block is stopped using sys.exit("done").
After all the sevel iterations are run, the resulting data structures ar
e saved onto a csv file and graphed below.
```

Python Program

In [58]:

```python
#Libraries
import numpy as np
from random import *
from pylab import *
```

In [59]:

```python
#Distance function - for calculating the distance between food and an ant's locat
def dis(a,b) :
    dx = a[0] - b[0]
    dy = a[1] - b[1]
    l = sqrt(dx**2 + dy**2)
    return l

#Here (at ants'colony), the strenth formula doesn't have any effect#
x = 50
y = 50
s = (10**(3.7112))/(((sqrt(((x)**2) + ((y)**2)))+1)**2)
print(log2(s))

#The strength of smell starts affecting the
#behavior of ants from here onwards. Before this point, ants' paths are complete
x = 35
y = 35
s = (10**(3.7112))/(((sqrt(((x)**2) + ((y)**2)))+1)**2)
print(int(log2(s)))

#This is the maximum strength of smell at food location
x = 0
y = 0
s = (10**(3.7112))/(((sqrt(((x)**2) + ((y)**2)))+1)**2)
print(int(log2(s)))
```

```
0.00010743550023507247
1
12
```

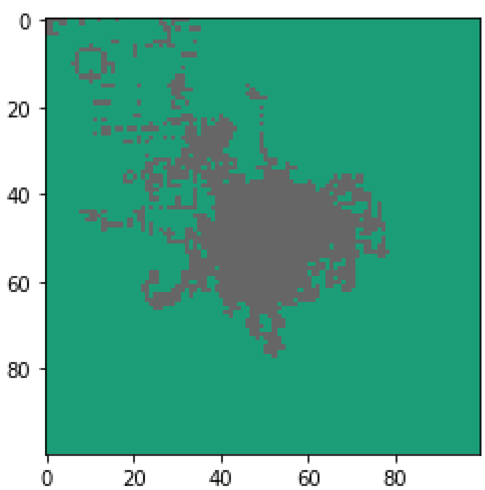
```
This value is not arbitrary as the furthest point from the food source i
s (1000,1000) and the strength of smell is closest to zerpo when s = (10
**6.3017)/((r+1)^2)
```

In [60]:
```
#Data Structures to store the time to reach the food source in each iteration
iteration = []
finish_time = []
```

In [61]:
```
#Code with sense of smell bias.
#No. of Ants(Na) = 100
#No Pheromone Bias involved
it = 1
Na = 100
nr = 100
nc = 100
#path_accumulation
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
#grid_update
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 4 and l >= 3 :
            C[i,j] = 1000
#current location of all ants
cur = array([50,50])
goal = dis(Food,cur)
gl = []
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            # Here, s is the strenth of smell at a particular point
            # which is inversely proportional to the #
            # distance from food source #
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
           # There's always a bias to move a bit more toward the left direction 
           #equal to (int(log2(s))) #
                temp = ic[i] - 1 - int(np.log(s))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
            # Same as above for moving up#
                temp = ir[i] - 1 - int(np.log(s))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,100)
           # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
```

```
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
                clf()
                imshow(Fa,cmap = 'Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
```

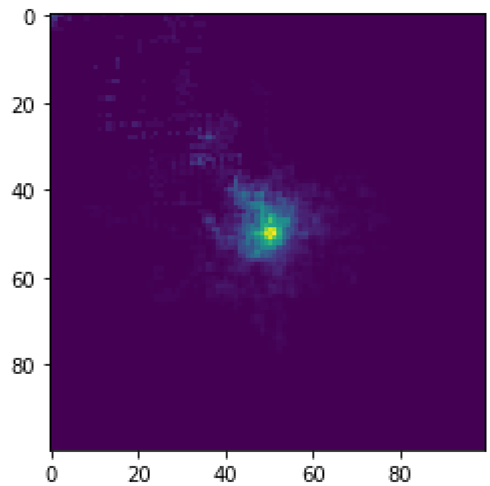
```
Distance from food = 2.0
Total Step = 166
```




```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```

In [62]:
```
#pheromone_update
pc = 0.0005      #pheromone_coefficient
Fz = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish -1]
    last = array([s[0],s[1]])
    phe = (1/(dis(Food,last)))*10
    for (kp,vp) in va.items() :
        Fz[vp[0],vp[1]] += phe/30
imshow(Fz)
```

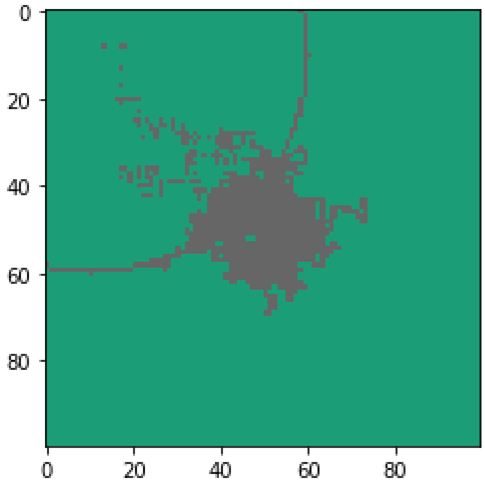
Out[62]: <matplotlib.image.AxesImage at 0x178d9b1bac8>



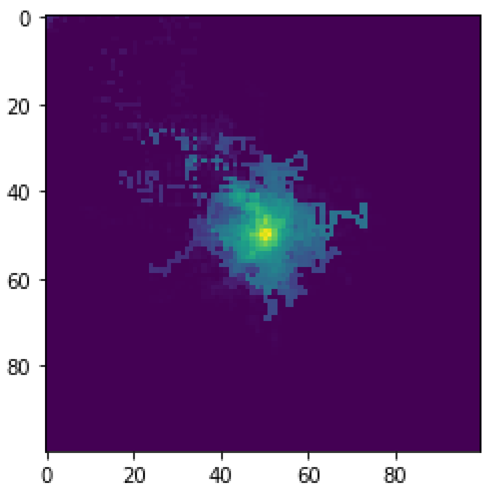
In [63]:
```
#Second Iteration
it = 2
Na = 100
nr = 100
nc = 100
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 50 and l >= 49 :
            C[i,j] = 1
cur = array([50,50])
goal = dis(Food,cur)
gl = []
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            ph = Fz[ir[min(max(i,0),99)],ic[min(max(i,0),99)]]
            o = s + ph
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
                temp = ic[i] - 1 - int(np.log(o))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
                temp = ir[i] - 1 - int(np.log(o))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,99)
          # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
                clf()
```

```
                imshow(Fa ,cmap ='Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                #pheromone_update
                tempMat = zeros((100,100))
                for (ka,va) in A.items() :
                    s = va[finish-1]
                    last = array([s[0],s[1]])
                    phe = (1/dis(Food,last))*10
                    for (kp,vp) in va.items() :
                        tempMat[vp[0],vp[1]] = 1
                    for i in range(100) :
                        for j in range(100) :
                            if tempMat[i,j] == 1 :
                                Fz[i,j] +=  phe/30
                            Fz[i,j] *= (1-pc)
                imshow(Fz)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
#pheromone_update
tempMat = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish-1]
    last = array([s[0],s[1]])
    phe = (1/dis(Food,last))*10
    for (kp,vp) in va.items() :
        tempMat[vp[0],vp[1]] = 1
    for i in range(100) :
        for j in range(100) :
            if tempMat[i,j] == 1 :
                Fz[i,j] +=  phe/30
            Fz[i,j] *= (1-pc)
imshow(Fz)
```

```
Distance from food = 3.605551275463989
Total Step = 98
```




```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```

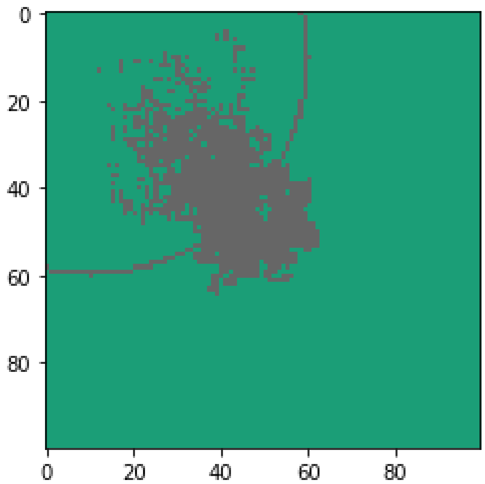


In [64]:

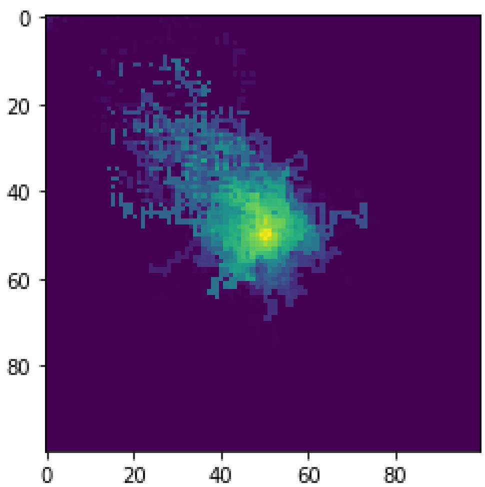
```
#Third Iteration
it = 3
Na = 100
nr = 100
nc = 100
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 50 and l >= 49 :
            C[i,j] = 1
cur = array([50,50])
goal = dis(Food,cur)
gl = []
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            ph = Fz[ir[min(max(i,0),99)],ic[min(max(i,0),99)]]
            o = s + ph
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
                temp = ic[i] - 1 - int(np.log(o))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
                temp = ir[i] - 1 - int(np.log(o))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,99)
          # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
                clf()
```

```
                imshow(Fa ,cmap ='Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                #pheromone_update
                tempMat = zeros((100,100))
                for (ka,va) in A.items() :
                    s = va[finish-1]
                    last = array([s[0],s[1]])
                    phe = (1/dis(Food,last))*10
                    for (kp,vp) in va.items() :
                        tempMat[vp[0],vp[1]] = 1
                    for i in range(100) :
                        for j in range(100) :
                            if tempMat[i,j] == 1 :
                                Fz[i,j] +=  phe/30
                            Fz[i,j] *= (1-pc)
                imshow(Fz)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
#pheromone_update
tempMat = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish-1]
    last = array([s[0],s[1]])
    phe = (1/dis(Food,last))*10
    for (kp,vp) in va.items() :
        tempMat[vp[0],vp[1]] = 1
    for i in range(100) :
        for j in range(100) :
            if tempMat[i,j] == 1 :
                Fz[i,j] +=  phe/30
            Fz[i,j] *= (1-pc)
imshow(Fz)
```

```
Distance from food = 3.605551275463989
Total Step = 79
```




```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```

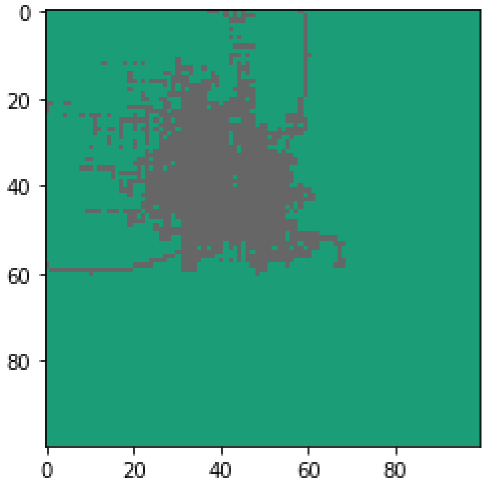


In [65]:

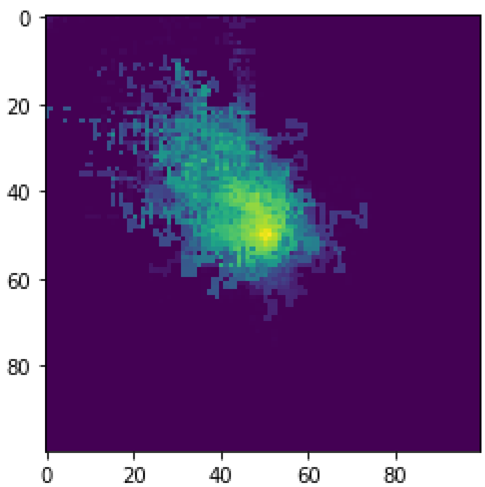
```
#Fourth Iteration
it = 4
Na = 100
nr = 100
nc = 100
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 50 and l >= 49 :
            C[i,j] = 1
cur = array([50,50])
goal = dis(Food,cur)
gl = []
imshow(Fa ,cmap ='Dark2')
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            ph = Fz[ir[min(max(i,0),99)],ic[min(max(i,0),99)]]
            o = s + ph
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
                temp = ic[i] - 1 - int(np.log(o))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
                temp = ir[i] - 1 - int(np.log(o))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,99)
           # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
```

```
                clf()
                imshow(Fa ,cmap ='Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                #pheromone_update
                tempMat = zeros((100,100))
                for (ka,va) in A.items() :
                    s = va[finish-1]
                    last = array([s[0],s[1]])
                    phe = (1/dis(Food,last))*10
                    for (kp,vp) in va.items() :
                        tempMat[vp[0],vp[1]] = 1
                    for i in range(100) :
                        for j in range(100) :
                            if tempMat[i,j] == 1 :
                                Fz[i,j] +=  phe/30
                            Fz[i,j] *= (1-pc)
                imshow(Fz)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
#pheromone_update
tempMat = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish-1]
    last = array([s[0],s[1]])
    phe = (1/dis(Food,last))*10
    for (kp,vp) in va.items() :
        tempMat[vp[0],vp[1]] = 1
    for i in range(100) :
        for j in range(100) :
            if tempMat[i,j] == 1 :
                Fz[i,j] +=  phe/30
            Fz[i,j] *= (1-pc)
imshow(Fz)
```

```
Distance from food = 3.605551275463989
Total Step = 76
```



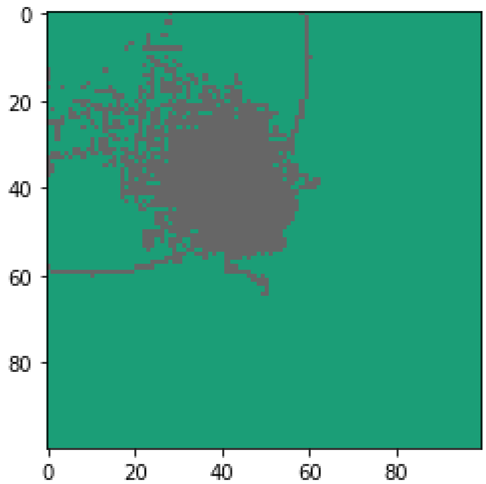

```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```



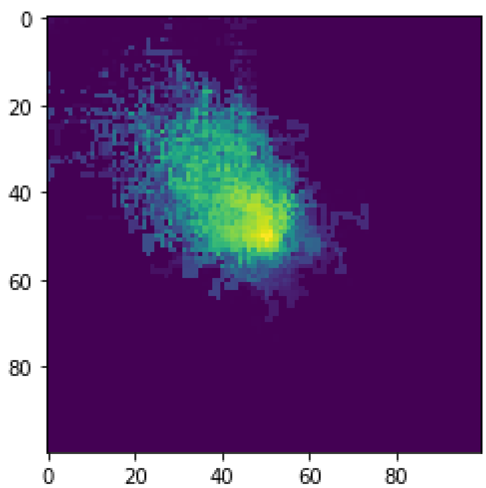
In [66]:
```
#Fifth Iteration
it = 5
Na = 100
nr = 100
nc = 100
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 50 and l >= 49 :
            C[i,j] = 1
cur = array([50,50])
goal = dis(Food,cur)
gl = []
imshow(Fa ,cmap ='Dark2')
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            ph = Fz[ir[min(max(i,0),99)],ic[min(max(i,0),99)]]
            o = s + ph
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
                temp = ic[i] - 1 - int(np.log(o))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
                temp = ir[i] - 1 - int(np.log(o))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,99)
           # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
```

```
                clf()
                imshow(Fa ,cmap ='Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                #pheromone_update
                tempMat = zeros((100,100))
                for (ka,va) in A.items() :
                    s = va[finish-1]
                    last = array([s[0],s[1]])
                    phe = (1/dis(Food,last))*10
                    for (kp,vp) in va.items() :
                        tempMat[vp[0],vp[1]] = 1
                    for i in range(100) :
                        for j in range(100) :
                            if tempMat[i,j] == 1 :
                                Fz[i,j] +=  phe/30
                            Fz[i,j] *= (1-pc)
                imshow(Fz)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
#pheromone_update
tempMat = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish-1]
    last = array([s[0],s[1]])
    phe = (1/dis(Food,last))*10
    for (kp,vp) in va.items() :
        tempMat[vp[0],vp[1]] = 1
    for i in range(100) :
        for j in range(100) :
            if tempMat[i,j] == 1 :
                Fz[i,j] +=  phe/30
            Fz[i,j] *= (1-pc)
imshow(Fz)
```

```
Distance from food = 4.242640687119285
Total Step = 70
```



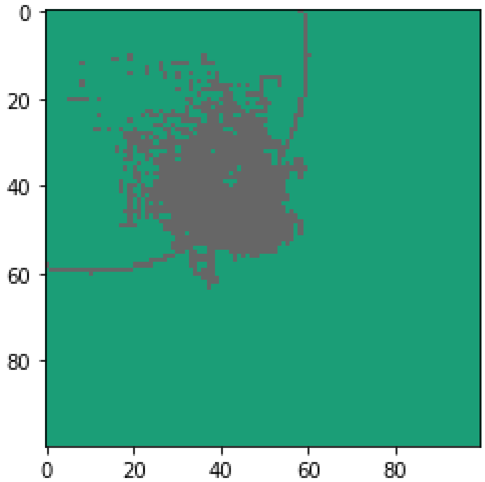

```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```



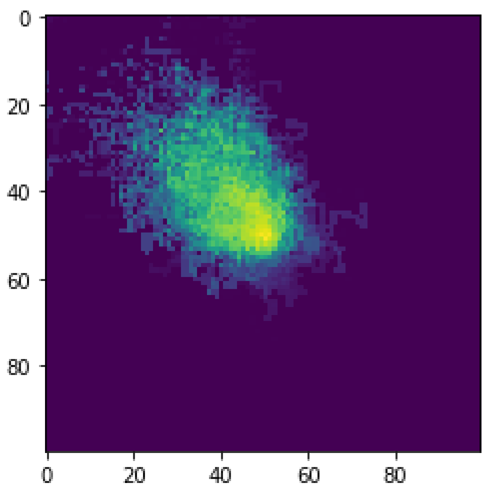
In [67]:
```
#Sixth Iteration
it = 6
Na = 100
nr = 100
nc = 100
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 50 and l >= 49 :
            C[i,j] = 1
cur = array([50,50])
goal = dis(Food,cur)
gl = []
imshow(Fa ,cmap ='Dark2')
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            ph = Fz[ir[min(max(i,0),99)],ic[min(max(i,0),99)]]
            o = s + ph
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
                temp = ic[i] - 1 - int(np.log(o))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
                temp = ir[i] - 1 - int(np.log(o))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,99)
           # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
```

```
                clf()
                imshow(Fa ,cmap ='Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                #pheromone_update
                tempMat = zeros((100,100))
                for (ka,va) in A.items() :
                    s = va[finish-1]
                    last = array([s[0],s[1]])
                    phe = (1/dis(Food,last))*10
                    for (kp,vp) in va.items() :
                        tempMat[vp[0],vp[1]] = 1
                    for i in range(100) :
                        for j in range(100) :
                            if tempMat[i,j] == 1 :
                                Fz[i,j] +=  phe/30
                            Fz[i,j] *= (1-pc)
                imshow(Fz)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
#pheromone_update
tempMat = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish-1]
    last = array([s[0],s[1]])
    phe = (1/dis(Food,last))*10
    for (kp,vp) in va.items() :
        tempMat[vp[0],vp[1]] = 1
    for i in range(100) :
        for j in range(100) :
            if tempMat[i,j] == 1 :
                Fz[i,j] +=  phe/30
            Fz[i,j] *= (1-pc)
imshow(Fz)
```

```
Distance from food = 2.8284271247461903
Total Step = 63
```



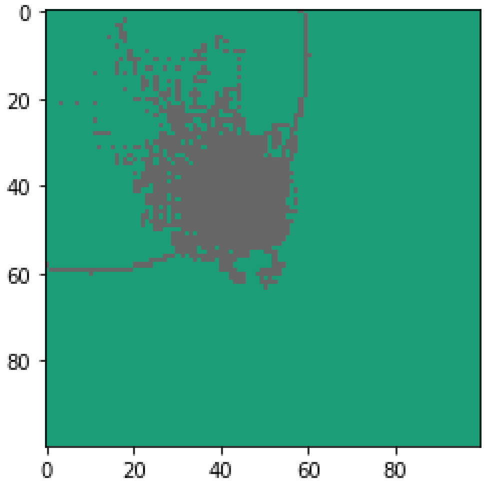

```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```



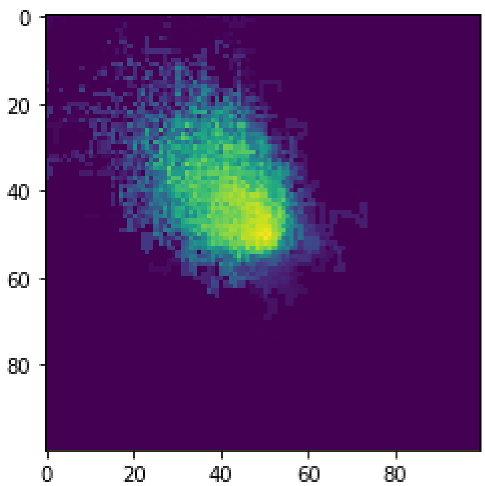
In [68]:
```
#Seventh Iteration
it = 7
Na = 100
nr = 100
nc = 100
A = {}
ants = linspace(0,99,100)
for i in ants:
    Path = {}
    A[int(i)] = Path
Fa = zeros((nr,nc))
ir = [int(nr/2)]*Na
ic = [int(nc/2)]*Na
Food = array([10,10])
C = zeros((nr,nc))
for i in range(nr) :
    for j in range(nc) :
        a = array([i,j])
        l= dis(Food,a)
        if l <= 50 and l >= 49 :
            C[i,j] = 1
cur = array([50,50])
goal = dis(Food,cur)
gl = []
imshow(Fa ,cmap ='Dark2')
for j in range(500):
        for i in range(Na) :
            p = rand(1)
            s = (10**(3.71117))/(((sqrt(((ic[i]-10)**2) + ((ir[i]-10)**2)))+1)**
            ph = Fz[ir[min(max(i,0),99)],ic[min(max(i,0),99)]]
            o = s + ph
            if p <= 0.25 :
                temp = ic[i] + 1
                ic[i] = min(temp,99)
            elif p>0.25 and p <= 0.5 :
                temp = ic[i] - 1 - int(np.log(o))
                ic[i] = max(temp,0)
            elif p>0.5 and p<=0.75 :
                temp = ir[i] - 1 - int(np.log(o))
                ir[i] = max(temp,0)
            else :
                temp = ir[i] + 1
                ir[i] = min(temp,99)
           # There's no further incentive, except by the random toss, to move do
            Fa[ir[i],ic[i]] = i
            Fa = Fa + C
            Fa[Fa>0] = 1
            temp = (ir[i],ic[i])
            A[i][j] = temp
            cur = array([ir[i],ic[i]])
            goal = dis(Food,cur)
            gl.append(goal)
            if goal < 5 :
                print("Distance from food = " + str(min(gl)))
                finish = j
                print("Total Step = " + str(finish))
```

```
                clf()
                imshow(Fa ,cmap ='Dark2')
                pause(0.01)
                iteration.append(it)
                finish_time.append(finish)
                #pheromone_update
                tempMat = zeros((100,100))
                for (ka,va) in A.items() :
                    s = va[finish-1]
                    last = array([s[0],s[1]])
                    phe = (1/dis(Food,last))*10
                    for (kp,vp) in va.items() :
                        tempMat[vp[0],vp[1]] = 1
                    for i in range(100) :
                        for j in range(100) :
                            if tempMat[i,j] == 1 :
                                Fz[i,j] +=  phe/30
                            Fz[i,j] *= (1-pc)
                imshow(Fz)
                sys.exit("done")
print("Distance from food = " + str(min(gl)))
print("Total Step = " + str(finish))
clf()
imshow(Fa ,cmap ='Dark2')
pause(0.01)
iteration.append(it)
finish_time.append(finish)
#pheromone_update
tempMat = zeros((100,100))
for (ka,va) in A.items() :
    s = va[finish-1]
    last = array([s[0],s[1]])
    phe = (1/dis(Food,last))*10
    for (kp,vp) in va.items() :
        tempMat[vp[0],vp[1]] = 1
    for i in range(100) :
        for j in range(100) :
            if tempMat[i,j] == 1 :
                Fz[i,j] +=  phe/30
            Fz[i,j] *= (1-pc)
imshow(Fz)
```

```
Distance from food = 4.123105625617661
Total Step = 59
```




```
An exception has occurred, use %tb to see the full traceback.

SystemExit: done
```



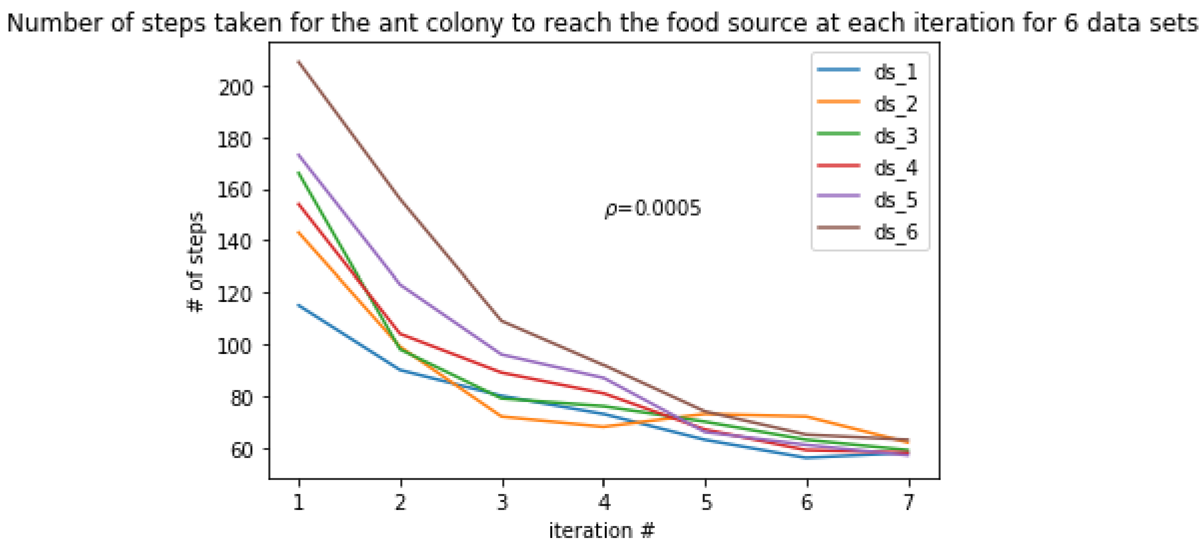
In [ ]:
```
print(iteration); print(finish_time)
```

In [69]:
```
import csv
with open('pheromone.csv', 'a', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(finish_time)
```

In [71]:
```
finish_times = []
with open('pheromone.csv', 'r', newline='') as file:
    reader = csv.reader(file)
    for row in reader :
        ind_time = []
        for f in row :
            ind_time.append(int(f))
        finish_times.append(ind_time)
print(finish_times)
```

```
[[115, 90, 80, 73, 63, 56, 58], [143, 99, 72, 68, 73, 72, 62], [166, 98, 79, 7
6, 70, 63, 59], [154, 104, 89, 81, 67, 59, 58], [173, 123, 96, 87, 66, 61, 57],
[209, 156, 109, 92, 74, 65, 63]]
```

In [88]:
```
for i in finish_times :
    plot(iteration, i)
xlabel('iteration #');ylabel('# of steps'); title('Number of steps taken for the
text(4,150, r'$\rho$=', fontsize=10)
text(4.3,150, 0.0005, fontsize = 10)
legend(('ds_1','ds_2','ds_3','ds_4','ds_5','ds_6'))
show()
```



Discussion

The result confirms that the number of steps it takes for the first ant in a colony to reach a food source clearly decreases with the number of iteration. Moreover, the data suggests that the difference in the 1st it eration(no pheromone trail) and the 2nd iteration(1st pheromone trail) i s significant. Furthermore, the graph demonstrates that at least a local optima has been found, and hence, the pheromone evaporation coefficient is efficient in providing a mechanism for selecting shorter paths. Ther efore, in line with the previous algorithms, an artificial pheromone sys tem is highly effective in optimizing the number of steps it takes for a n ant colony to find food.

However, it is beyond the scope of this algorithm to accurately simulate ant-foraging patterns. Nonetheless, the algorithm provides a near-comprehensive picture of how ant colonies tackle the problem of path-building and food-foraging.
An important implication of this result is that it demonstrates how a seemingly unsophisticated colony of ants traverse paths together to reach a food source and are able to optimize the route just by using small dozes of secreted chemicals.

Conclusion

The algorithm aimed to simulate an ant-foraging pattern with artificial pheromone system. Based on the line graph of the number of steps over the number of iterations, we can conclude that pheromone trails serve as the experience-factor in the food-foraging patterns found in ants. The results indicate that a pheromone system exponentially decreases the number of steps it takes an ant colony to reach a food location.