Solution for Optiver's puzzle

Jialing Yu

January 2024

Puzzle's statement

Q: An ant leaves its anthill in order to forage for food. It moves with the speed of 10cm per second, but it doesn't know where to go, therefore every second it moves randomly 10cm directly north, south, east or west with equal probability.

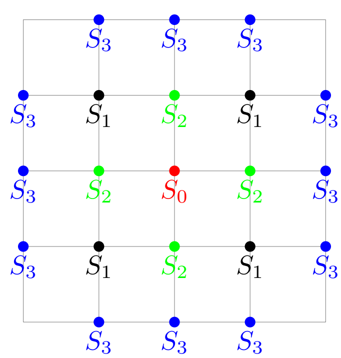
1 Q1: 4.5 seconds

Q1: If the food is located on east-west lines 20cm to the north and 20cm to the south, as well as on north-south lines 20cm to the east and 20cm to the west from the anthill, how long will it take the ant to reach it on average?

A: The average time is 4.5 seconds.

1.1 Solution

We can model the problem into a markov chain to solve the problem. The markov chain has 4 states s_0, s_1, s_2, s_3 which can be drawn as follow:



s_0 is the start point of the ant, s_1 are the possible location of the ant after its first step, s_2 are the possible locations of the ant after the second step, s_3 are the possible locations of the ant after the third step, which are also the absorption states(i.e., the locations of the food).

Denote t_i to be the expected time from state s_i to absorption state s_3, obviously

$$t_3 = 0$$

since the ant is already at the location of food, our goal is to compute t_0, which is the expected time for the ant from the start point to the food. Since the ant start at s_0 and have equal probability going any direction, after the first step, the ant must be in state s_1 and we have

$$t_0 = 1 + t_1$$

; if the ant at state s_1, after one step the ant has 1/4 probability in state s_3, 1/2 probability in state s_2 and 1/4 probability in state s_0, thus we have

$$t_1 = 1 + \frac{1}{4}t_3 + \frac{1}{2}t_2 + \frac{1}{4}t_0$$

; if the ant at state s_2, after one step, the ant has 1/2 probability at state s_3, and 1/2 probability at state s_1, thus

$$t_2 = 1 + \frac{1}{2}t_3 + \frac{1}{2}t_1.$$

Solving the above equations we have

$$t_0 = 4.5.$$

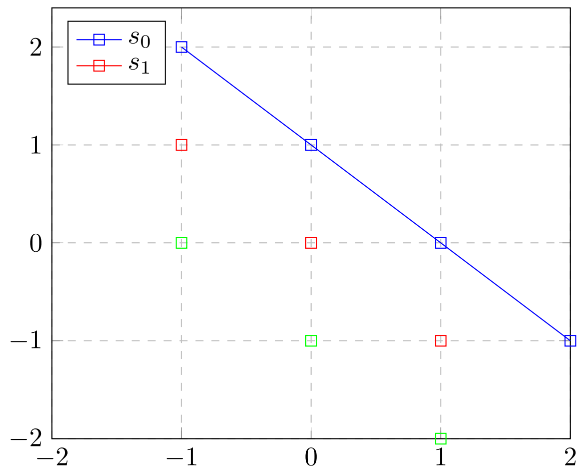
2 Q2: ∞

Q2: What is the average time the ant will reach food if it is located only on a diagonal line passing through (10cm, 0cm) and (0cm, 10cm) points?

A: The average time is ∞.

2.1 Solution

Denote s_0 to be the locations of food, s_1 to be the locations $1cm$ on the left of the food horizontally and $1cm$ under the food vertically; similarly for s_n.



The ant start at s_1 and we want to compute the average time for it to reach s_0. Denote t_i to be the average time from state s_i to s_0, we want to compute t_1. The problem can be modeled as a 1-D random walk: set the start point s_1 to be the integer 1 of the 1-D coordinate, after one step, the ant has equal probability be in s_0(1 at the 1-D coordinate) or s_2(2 at the 1-D coordinate), thus we have

$$t_1 = 1 + \frac{1}{2}t_0 + \frac{1}{2}t_2$$

; if the ant were at s_0, it stop, thus

$$t_0 = 0$$

, if the ant were at s_2, after one step, it again has equal probability be at s_1 or s_3, thus we have

$$t_2 = 1 + \frac{1}{2}t_1 + \frac{1}{2}t_3$$

; apply the same relation to all n, we have

$$t_n = 1 + \frac{1}{2}t_{n-1} + \frac{1}{2}t_{n+1}$$

, which is the same as

$$(t_n - t_{n-1}) - (t_{n+1} - t_n) = 2$$

.

Solve for t_1 by using the above relationship, we have $t_n = n(t_1 - n + 1)$. Since when $n \to \infty$, t_n must be ∞, we have $t_1 = \infty$.

Or we can think of it as when the ant starts at the integer 1 of the $1 - D$ coordinate, it has equal probability of going left or right, thus after one step, the average position of the ant is $1 + \frac{1}{2} \times 1 + \frac{1}{2} \times (-1) = 1$, i.e., on average, the ant would not change its position, thus the average time for reaching integer 0 is ∞.

3 Q3: 14 seconds

Q: Can you write a program that comes up with an estimate of average time to find food for any closed boundary around the anthill? What would be the answer if food is located outside an area defined by $((x-2.5cm)/30cm)^2+((y-2.5cm)/40cm)^2 < 1$ in coordinate system where the anthill is located at (x = 0cm, y = 0cm)?

A. The average time is 14 seconds.

3.1 Solution

We can simulate the process as follow for 100000 times, and the average time is around 14 seconds. For any general closed curve, we can simply change the formulation for f in the code.

```
import numpy as np

# define a closed curve
def f(x,y):
    return  ( (x - 2.5) / 30 )**2 + ( (y - 2.5) / 40 )**2 - 1


# simulate the process 100000 times
step_sum = 0
for i in range(100000):
    x,y,step = 0,0,0
    # while the ant (x,y) is inside the closed curve
    # the ant moves in 4 directions with the same probabilities
    # repeat the process until the ant go out of the curve
    while f(x,y) <0:
        direction = np.random.randint(low = 0, high =4)
        if direction == 0:
            x += 10
        if direction == 1:
            x -= 10
        if direction == 2:
            y += 10
        if direction == 3:
            y -= 10
        # count the number of total steps
        step += 1
    step_sum += step

step_sum/100000

13.98287
```

Listing 1: simulation of the process