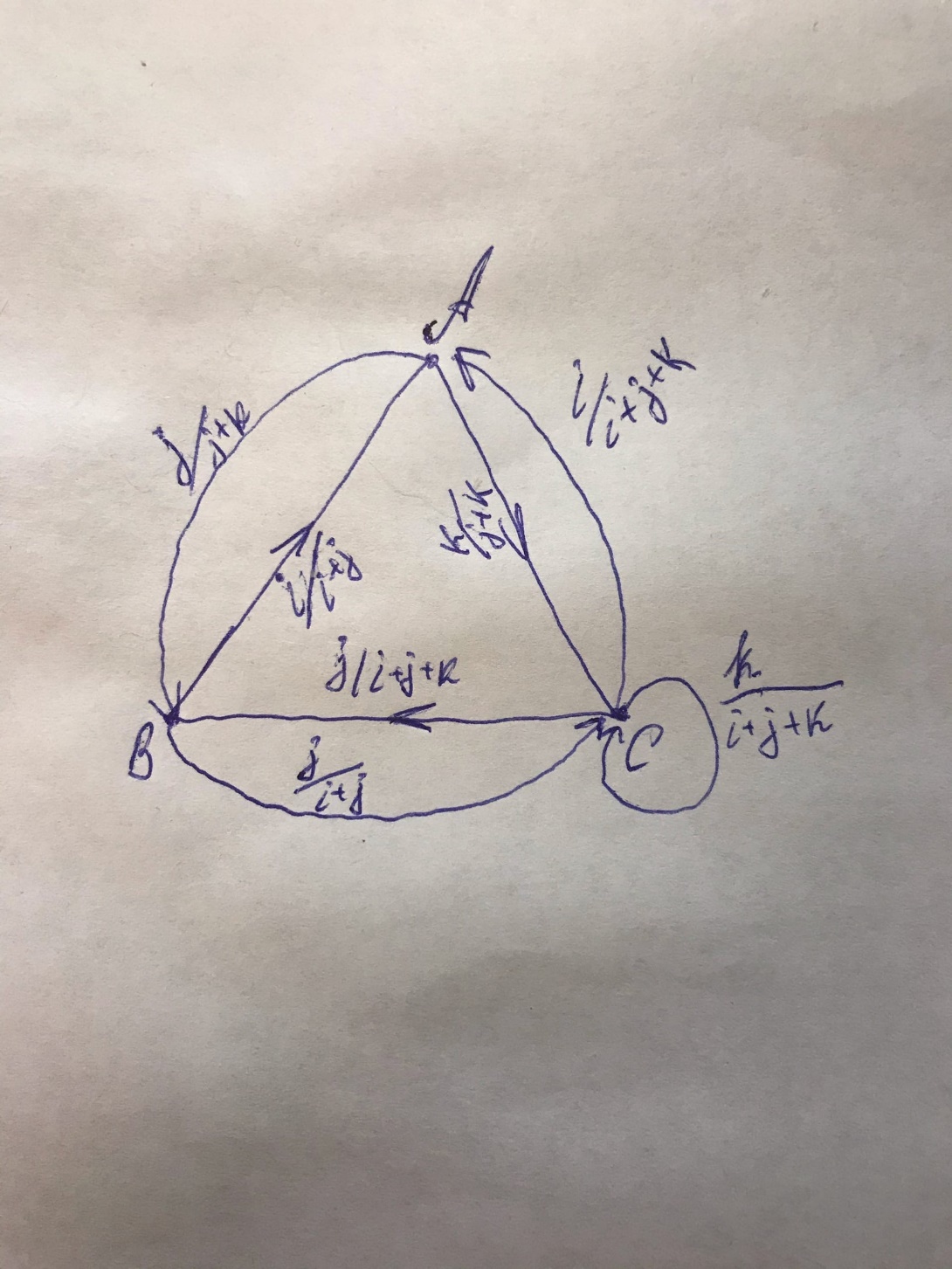
Лабораторная работа №3

«Эргодические марковские цепи»

Задача о мухе

Имеется треугольник с вершинами A, B, C. В начальный момент времени муха находится на вершине A. Каждую секунду муха перелетает на другую вершину или на ту же самую. Необходимо определить через какое время муха вернётся в начальную вершину.



Исходные данные варианта

Ход работы

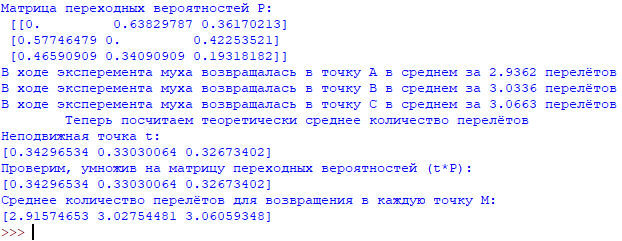
1. Получим матрицу переходных вероятностей:

1. Проведём эксперимент и выясним, за какое среднее время муха будет возвращаться в исходную точку. В ходе 10000 испытаний муха возвращалась в среднем за 2.92 перелёта.
2. Найдём непрерывную точку.
3. Проверим правильность нахождения непрерывной точки, для этого умножим её на матрицу переходных вероятностей:

Вектор t не изменился, значит является неподвижной точкой.

1. Среднее время возвращения теоретически:

Пример работы программы



Код программы

import random

import math

import numpy

import matplotlib.pyplot as pyplot

from numpy import linalg as la

i\_ = 41

j\_ = 30

k\_ = 17

sz = 10000 #количество испытаний

ab = j\_ / (j\_ + k\_)

ac = k\_ / (j\_ + k\_)

ba = i\_ / (i\_ + j\_)

bc = j\_ / (i\_ + j\_)

ca = i\_ / (i\_ + j\_ + k\_)

cb = j\_ / (i\_ + j\_ + k\_)

cc = k\_ / (i\_ + j\_ + k\_)

# Матрица переходных вероятностей

P = numpy.array([

[0, ab, ac], # A

[ba, 0, bc], # B

[ca, cb, cc]]) # C

# A B C

print("Матрица переходных вероятностей P:\n", P)

for k in range (0, 3):

state = base\_state = k #состояние: 0 - A, 1 - B, 2 - C

count = 0 # количество перелётов

for i in range(0, sz):

r = random.random()

j = 0

while r > P[state][j]:

r -= P[state][j]

j += 1

state = j

count += 1

while state != base\_state:

r = random.random()

j = 0

while r > P[state][j]:

r -= P[state][j]

j += 1

state = j

count += 1

count /= sz

print("В ходе эксперемента муха возвращалась в точку {} в среднем за {} перелётов".format(chr(65+k), count))

print("\tТеперь посчитаем теоретически среднее количество перелётов")

#t - непдвижная точка, P1 - матрица коэффициетов СЛАУ

P1 = [[-1, ba, ca],

[ab, -1, cb],

[1, 1, 1]]

b = [0, 0, 1] #столбец свободных членов

t = la.solve(P1, b)

print("Неподвижная точка t:")

print(t)

print("Проверим, умножив на матрицу переходных вероятностей (t\*P):")

#print(numpy.dot(P, t))

print((numpy.transpose(P).dot(t)))

M = 1/t

print("Среднее количество перелётов для возвращения в каждую точку M:")

print(M)