



دانشگاه شهید بهشتی
دانشکده فیزیک

گزارش کار برنامه شبیه‌سازی انتقال بیماری

محدثه مظفری

آبان ۱۴۰۲

۱ مقدمه

این برنامه یک مدل ساده از انتقال بیماری در یک جمعیت با استفاده از یک حلقه ارائه می‌دهد. برنامه با استفاده از متغیرهای ALPHA، BETA، NUM_ILLNESS و STEPS تنظیمات اجرای شبیه‌سازی را مشخص می‌کند.

با گذر زمان (STEPS)، انتقال بیماری بر اساس ALPHA و BETA اعمال می‌شود. اگر یک فرد بیمار باشد، با احتمال ALPHA به افراد همسایه بیماری منتقل می‌شود. با احتمال BETA فرد بیمار به حالت ۲ (بهبود یافته) تغییر می‌کند. حالت نهایی جمعیت پس از گذر از تمام مراحل نمایش داده می‌شود.

۲ نتایج شبیه‌سازی

برای شرایط اولیه $N = 10$ ، $ALPHA = 0.3$ ، $BETA = 0.2$ ، $NUM_ILLNESS = 2$ و $STEPS = 5000$ نتیجه شبیه‌سازی به صورت زیر می‌باشد:

خروجی کد
0101000000
2222222222

حالت اول شرایط اولیه را نمایش می‌دهد. که اعداد ۱ نشانگر افراد بیمار و اعداد ۰ نشانگر افراد سالم است. حالت دوم، حالت نهایی سیستم را نشان می‌دهد که همه افراد بهبودیافته اند. برای شرایط اولیه $N = 10$ ، $ALPHA = 0.2$ ، $BETA = 0.2$ ، $NUM_ILLNESS = 2$ و $STEPS = 5000$ نتیجه شبیه‌سازی به صورت زیر می‌باشد:

خروجی کد
0000100010
2222200022

همان‌طور که مشاهده می‌شود؛ با تغییر مقدار ALPHA حالت نهایی سیستم تغییر می‌کند. زیرا ممکن است افراد بهبودیافته باشند و دیگر نتوانند همه سیستم را بیمار کنند.

۳ کد شبیه‌سازی

```
۱ #include <iostream>
۲ #include <vector>
```

```

3 #include <random>
4
5 using namespace std;
6
7 void initial_condition(vector<int>& ring, int num, mt19937& gen,
8     uniform_int_distribution<int> ran_pos){
9     for (int i = 0; i < num; i++)
10     {
11         ring[ran_pos(gen)] = 1;
12     }
13 }
14
15 int main(){
16     const int SIZE = 10;
17     const double ALPHA = 0.2;
18     const double BETA = 0.2;
19     const int NUM_ILLNESS = 2;
20     const int STEPS = 5000;
21
22     mt19937 gen(random_device{}());
23     uniform_int_distribution<int> ran_pos(0, SIZE-1); //Get any random
24     integer
25     uniform_real_distribution<double> ran_u(0.0, 1.0); //Our uniform
26     variable generator
27
28     vector<int> ring(SIZE, 0);
29     initial_condition(ring, NUM_ILLNESS, gen, ran_pos);
30     //initial state
31     for (size_t i = 0; i < ring.size(); i++){
32         cout<<ring[i];
33     }
34     cout << endl;
35
36     for (int t = 0; t < STEPS; t++){
37         vector<int> illness_index;
38         for (size_t i = 0; i < ring.size(); i++){
39             if (ring[i] == 1){
40                 illness_index.push_back(i);

```

```

39     }
40 }
41 for (const auto i :illness_index){
42     if (ran_u(gen) < ALPHA){
43         if (ring[(i+1) % SIZE] != 2){
44             ring[(i+1) % SIZE] = 1;
45         }
46
47     }
48     if (ran_u(gen) < ALPHA){
49         if (ring[(i-1+SIZE) % SIZE] != 2){
50             ring[(i-1+SIZE) % SIZE] = 1;
51         }
52     }
53     if (ran_u(gen) < BETA){
54         ring[i] = 2;
55     }
56
57 }
58 }
59 for (size_t i = 0; i < ring.size(); i++){
60     cout<<ring[i];
61 }
62 return 0;
63 }

```