河南大学2022~2023学年第2学期

数学建模（校通识课）

课程论文

|  |  |  |  |
| --- | --- | --- | --- |
|  | 姓名 | 学号 | 学院 |
| 组长 | 王丽竹 | 2209070233 | 商学院 |
| 组员 | 杨馥瑞 | 2212080042 | 计算机与信息工程学院 |
| 组员 | 张莎莎 | 2208080292 | 经济学院 |

2023年6月14日

新型冠状病毒肺炎传播情况简易可视化模拟模型

摘要：新型冠状病毒肺炎，简称“新冠肺炎”，在2019年大范围传播，对全球人类健康造成巨大危害，世界经济遭受重大冲击，为了将其快速控制，对新冠肺炎的预测与控制变得十分重要。本文建立简单的传播模拟器，对疫情的传播情况进行可视化研究：针对本问题，我们选择利用元胞自动机方法进行模拟。其中，针对人口移动问题，我们首先使用两层循环在一确定范围内生成人口坐标，通过turtle库设置形状格式，再定义一个移动函数，随机选择方向，以上面坐标为基础进行修改，实现人口的移动可视化。针对新冠传播问题，需要定义一个传播函数，以便于检测附近人口是否被感染，计算被感染者与可能感染者距离，得出可能被感染者感染情况，同时将未被感染者从感染列表移除，得出最终疫情传播情况。针对被感染后续治疗与隔离问题，定义治疗与隔离函数，感染者在其中相应状态改为已进行，再根据生存天数判断是否从函数中移除或更改状态，得出感染者后续情况变化。最后针对新冠传播情况可视化模型建立问题，定义开始模拟函数，以天数为参数，对于未感染与已感染人口、已治愈人口和已隔离人口，调用相应函数操作，再定义统计函数，用于统计上面所提及人口，在每一天末输出各状态人口数，后利用turtle库进行可视化展示，得出可视化模型。

关键词：新型冠状病毒肺炎；元胞自动机模型；模拟

1. 基本假设

假设也是局限性：

（1）感染病毒后，人不会出现二次感染的情况。

（2）病毒的传播能力是固定的。且病毒不会出现变异。

（3）医院对每个人的治疗成功率都是固定的。

（4）病毒不存在自愈的可能，隔离在家。不接受医院治疗，只有死亡的结果。

（5）起始人口是十分的密集在一个固定区域的，且人们的移动能力不强。

（6）没有核酸检测等一系列预防措施。

（7）人们的移动是随机的，并不会趋利避害

（8）忽视了个体的差异性

二、元胞自动机模型

2.1模型简介

该模型是一个利用元胞自动机方法的简单的传染病传播模拟器，使用了turtle库进行可视化展示。它模拟了人口的移动、感染、治疗和死亡等过程，并提供了统计功能来分析传染病的传播情况。

2.2模型建立

2.2.1创建初始人口群体

for i in range(-personnumber, personnumber):  
 for j in range(-personnumber, personnumber):  
 individual = [[i, j], 100, 0, turtle.Turtle()] # [坐标, 状态（0：未感染，1：感染，2：已隔离，3：已死亡），天数，turtle对象]  
 population.append(individual) # 将人口添加到列表population中  
 turtle.delay(0) # 设置绘图延迟  
 individual[3].shape('circle') # 设置人口的形状  
 individual[3].resizemode('user') # 设置人口的缩放模式  
 individual[3].shapesize(0.3) # 设置人口的大小  
 individual[3].fillcolor('green') # 设置人口的颜色  
 individual[3].penup() # 抬起画笔，避免绘制轨迹  
 individual[3].goto((individual[0][0] \* 8) - 100, (individual[0][1] \* 8) - 100) # 将画笔移动到人口的初始位置  
  
movement\_directions = [[0, 1], [0, -1], [1, 0], [-1, 0], [0, 0]] # 定义人口移动的方向

使用两层循环在一个范围内生成人口的初始位置坐标。

每个人口表示为一个列表，包含位置坐标、状态（0：未感染，1：感染，2：已隔离，3：已死亡）、生存天数和turtle对象（用于可视化）。

通过turtle库设置人口的形状、大小和颜色，并将画笔移动到初始位置。

2.2.2人口移动功能

def move\_individual(individual):  
 individual[0][0] += movement\_directions[random.randint(0, 4)][0] # 随机选择一个方向并移动  
 individual[0][1] += movement\_directions[random.randint(0, 4)][1]  
  
 individual[3].goto((individual[0][0] \* 8) - 100, (individual[0][1] \* 8) - 100) # 将画笔移动到人口的新位置  
 return individual

定义一个移动函数，随机选择一个方向并移动人口的位置。

通过修改turtle对象的坐标和形状，实现人口的移动可视化。

2.2.3传染病传播功能

infection\_list = []  
for person in population:  
 if (person[0][0] - individual[0][0]) \*\* 2 + (person[0][1] - individual[0][1]) \*\* 2 < size and person[2] == 0: # 判断距离是否小于2，并且未感染  
 infection\_list.append(person) # 将可传染的人口添加到列表中  
 else:  
 continue  
for \_ in range(infection\_rate):  
 if len(infection\_list) == 0: # 如果没有可传染的人口，则结束传染  
 break  
 else:  
 random\_index = random.randint(0, (len(infection\_list) - 1)) # 随机选择一个可传染的人口进行传染  
 infection\_list[random\_index][2] = 1 # 将该人口状态设置为感染  
 infection\_list[random\_index][3].fillcolor('yellow') # 将该人口的颜色设置为黄色  
 infection\_list.remove(infection\_list[random\_index]) # 将该人口从可传染列表中移除

定义一个传播函数，用于检测附近的人口是否可以被感染。

通过计算两个人口之间的距离，判断是否小于2个单位距离，并且该人口尚未感染。

将可传染的人口添加到传染列表中，并随机选择一个进行感染。

修改感染人口的状态和颜色，并从传染列表中移除。

2.2.4治疗和隔离功能

def treat\_individual(individual):  
 individual[2] = 2 # 将病人状态设置为已治愈  
 if individual[1] >= 90: # 如果病人已经痊愈  
 population.remove(individual) # 将病人从总人口列表中移除  
 individual[3].reset() # 将病人的turtle对象清除  
 hospital\_capacity[1] += 1 # 将医院收治的人数加1  
 hospital\_capacity[0] -= 1 # 将医院剩余床位数减1  
 else: # 如果病人还没痊愈  
 individual[1] += death\_rate # 将病人的生存天数增加death\_rate  
 individual[3].fillcolor('blue') # 将病人的颜色设置为蓝色  
  
def isolate\_individual(individual): # 隔离  
 individual[2] = 3 # 将病人状态设置为已隔离  
 if individual[1] > 0: # 如果病人还活着  
 individual[1] -= recovery\_period # 将病人的生存天数减少recovery\_period  
 individual[3].fillcolor('red') # 将病人的颜色设置为红色  
 else: # 如果病人已经死亡  
 population.remove(individual) # 将病人从总人口列表中移除  
 num[0]+=1  
 individual[3].reset() # 将病人的turtle对象清除

定义治疗函数和隔离函数，用于对感染的人口进行治疗或隔离操作。

治疗函数将感染者的状态设置为已治愈，并根据生存天数判断是否需要从总人口中移除。

隔离函数将感染者的状态设置为已隔离，并根据生存天数判断是否需要将其移除或更改颜色。

2.3模拟过程

def start\_simulation(days):  
 for day in range(days):  
 for individual in population:  
 if individual[2] == 0 or individual[2] == 1: # 如果病人状态为未感染或者潜伏期  
 move\_individual(individual) # 对病人进行移动操作  
 elif individual[2] == 2: # 如果病人状态为已治愈  
 treat\_individual(individual) # 对病人进行治疗操作  
 else: # 如果病人状态为已隔离或者死亡  
 death\_operation(individual) # 对病人进行死亡操作  
 infected\_list = [x for x in population if x[2] == 1] # 找出所有处于潜伏期的病人  
 for infected in infected\_list:  
 spread\_infection(infected) # 对潜伏期病人进行感染操作  
 death\_operation(infected) # 对潜伏期病人进行死亡操作

定义一个开始模拟函数，接受天数作为参数。

在每一天的模拟中，遍历人口群体中的每个人口。

对于未感染和潜伏期的人口，调用移动函数进行移动操作。

对于已治愈和已隔离的人口，调用相应的治疗或死亡操作函数。

在每一天的模拟结束后，找出所有潜伏期的感染者，并对其进行感染和死亡操作。

模拟结束后，调用统计函数输出当前感染、未感染、死亡和治愈人数。

2.4统计功能

def statistics():  
 num\_infected = 0  
 num\_uninfected = 0  
 for individual in population:  
 if individual[2] == 1: # 统计当前感染人数  
 num\_infected += 1  
 elif individual[2] == 0: # 统计当前未感染人数  
 num\_uninfected += 1  
 else:  
 continue

定义统计函数，用于统计当前感染、未感染、死亡和治愈人数。

遍历人口群体中的每个人口，根据其状态进行计数。

这个模拟器使用turtle库进行可视化展示，通过移动、感染、治疗和死亡等功能，模拟了传染病的传播过程，并提供了统计功能来分析传染病的传播情况。

2.5参数修改

personnumber=int(input("请输入人口基数^2:"))  
bednumber=int(input("请输入病床数:"))  
size=float(input("请输入最大传播距离:"))  
cure=float(input("请输入治愈率:"))  
cricul=int(input("模拟天数:"))

并且，可以通过修改参数（医院床位数，医院的治疗率，模拟天数，传播距离，起始人数及距离）和调整功能来改变模拟的行为和结果。

2.6模型运行情况

模型已经打包成Window上的可执行程序，以便于与用户交互使用。

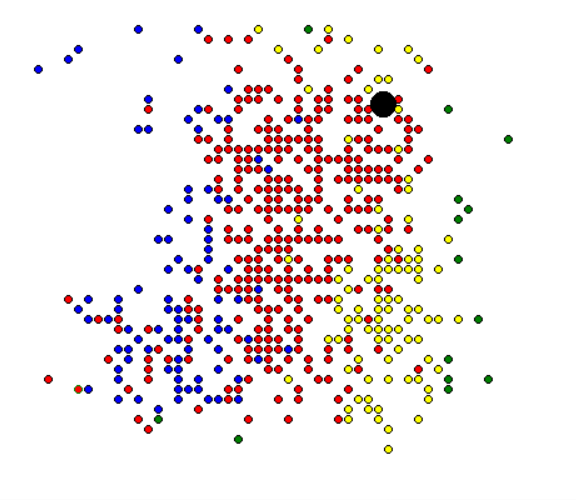
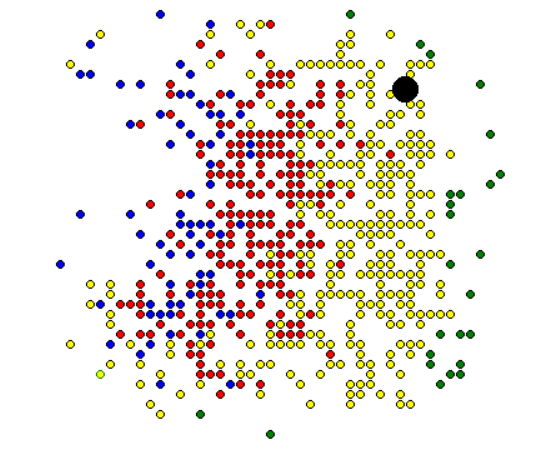
并且模型可以动态的实现感染情况的可视化，其中4种颜色代表四种情况。

（1）绿色：未感染。

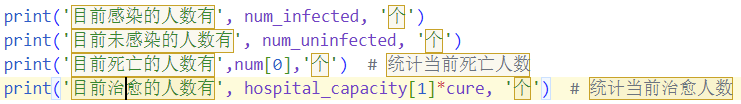
（2）黄色：潜伏期。

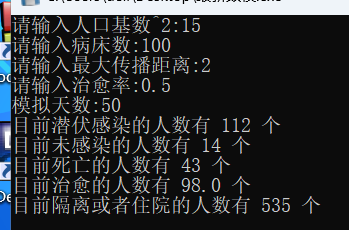
（3）红色：隔离。

（4）蓝色：住院治疗。



这是在不断模拟天数的其中两天的模拟情况。

并在模拟天数结束后，会将已经设置完的参数下得到的运行结果展示。



三、模型评价及改进

3.1模型评价

（1）模型的基本功能实现了人口的移动、感染、治疗和死亡等过程，以及统计功能，能够提供一定的传染病传播模拟效果。

（2）代码结构清晰，函数和变量命名合理，容易理解和修改。

（3）模型通过使用turtle库进行可视化展示，可以直观地观察人口的状态和传播过程。

3.2模型改进

1. 添加社会距离机制。可以限制在一定距离内的人口之间的传播,模拟社会疏散的效果。
2. 添加不同年龄段和职业的人口。不同年龄段和职业的人口传播率和死亡率不同,可以进行区分。
3. 添加医院收治机制。可以根据医院的床位数和病人的严重程度进行分流,较严重的病人优先入院治疗。
4. 添加隔离措施。可以在出现疫情后采取隔离措施,限制人口流动和聚集,观察其效果。
5. 添加政策干预。可以设置不同的政策,如限制城市封锁、停工停课等,观察其对疫情的影响。
6. 优化绘图。可以对人口的大小、颜色等进行优化,更加清晰地观察疫情的传播情况。
7. 优化参数。设置不同的参数,如传播率、死亡率、治愈率等,产生不同的模拟结果,对策略进行比较。
8. 统计更详细的数据。统计不同年龄段、职业的人口数量,以及感染者、死亡者、治愈者等数据,观察疫情对不同人群的影响。

这些改进可以使该模型更加全面和精细,可以用于评估不同的防疫策略,为政策制定提供参考。当然,模型的改进还需要不断优化,才能使其结果更加准确。

参考文献

[1]余雷,薛惠锋,高晓燕,等.基于元胞自动机的传染病传播模型研究[J].计算机工程与应用,2007,43(2):196-198,237

[2]游爱丽.基于元胞自动机的SIR传染病模型[D].新疆:新疆大学,2010.

[3]林俊锋.基于引入隐形传播者的SEIR模型的COVID-19疫情分析和预测[J/OL].电子科技大学学报:1-8.(2020-04-08)[2020-04-27].http://kns.cnki.net/kcms/detail/51.1207.T.20200408.1038.002.html.

[4]陈长坤,童蕴贺.基于元胞自动机的传染病跨区域传播模型研究[J].武汉理工大学学报:信息与管理工程版,2018,40(4):359-363,382.DOI:10.3963/j.issn.2095-3852.2018.04.001.

[5]吴成来.基于元胞自动机的传染病传播模型研究[J].电脑知识与技术,2019,15(09):185-186.