2022-2023学年冬季学期

《研究方法与前沿》(0830SY02)

课程报告

成绩

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学 号 | | 20123011 | 学 院 | 计算机科学与工程学院 | |
| 姓 名 | |  | 手工签名 |  | |
| 报告题目 | | SIS模型下的流行病传播过程仿真实践 | | | |
| 课程报告成绩50% | 报告主要部分：   1. 引言：对相关研究工作进行较全面的综述； 2. 报告内容：叙述条理清楚，有公式、算法描述； 3. 实验结果：结合图对结果分析，有结论； 4. 结论：有总结、分析、展望； 5. 附录：附核心代码，不粘贴代码，代码有注释或说明（40%） | | | |  |
| 书写格式：  书写规范、表述顺畅；图表、代码清晰规范，引用规范；  参考文献（不少于5篇参考文献，英文文献不少于2篇），引用格式规范。（10%） | | | |  |
| 工作实绩50% | 1、平时表现（25%）；2、工作实绩（25%） | | | |  |
| **报告评语：**  教师签名：  日期： 年 月 日 | | | | | |

SIS模型下的流行病传播过程仿真实践

# **摘 要**

# 新冠疫情席卷全球，SIS作为一种传统的传染病模型，能够帮助我们更好地了解流行病传播的规律，提出更有效的解决方案。本篇文章研究了SIS模型的解析解及其在复杂网络（小世界网络）下的传播规律，对上海开放后的疫情做了预测。同时，通过Netlogo软件对SIS模型进行了仿真，验证了保持一米距离对遏制传染病扩散的重要性。

# **关键词：SIS模型 流行病 新冠 仿真 小世界网络**

**Simulation practice of epidemic transmission process**

**under SIS model**

# **ABSTRACT**

**As the novel coronavirus epidemic has swept the world, SIS, as a traditional infectious disease model, can help us better understand the law of epidemic transmission and propose more effective solutions. In this paper, the analytical solution of SIS model and its propagation rule under complex network (small world network) are studied, and the epidemic situation in Shanghai after opening up is predicted. At the same time, the SIS model was simulated by Netlogo software, which verified the importance of keeping one meter distance to contain the spread of infectious diseases.**

**Key words: SIS model pandemic Covid-19 simulation Small world network**

# 引言

近年来，新冠疫情席卷全球，各个国家的经济遭到了打击，人民的生命安全遭到了威胁，新冠疫情已经与我们的生活密不可分。虽然我们不能快速消灭新冠，但我们可以用现有的科技来了解它、分析它、预测它，从而找出有效的解决方案来遏制它的传播，保护更多人的生命安全。

随着计算机科学的不断发展，跨学科的交流与融合逐渐成为一种常态。尤其是随着网络科学的发展，掀起了一个从复杂网络的角度研究传染病模型的热潮。而SIS模型是研究流行病传播规律的传统模型之一，它也是对传染病进行研究的重要手段[3]。通过分析新冠疫情在SIS模型下的传播，我们能够在一定程度上了解新冠的传播规律，给出一些行之有效的预防措施。

本文将从解析解、Netlogo仿真、数值拟合、小世界网络等多个角度研究、分析新冠疫情在SIS模型下的传播规律，寻找出影响新冠疫情传播的关键因素，探讨出遏制新冠疫情的有效方法，为新冠疫情的防治决策提供科学依据。同时，本文还将预测在SIS模型下2022年12月13日全面放开对上海疫情的影响。

# 模型及研究方法

## 模型介绍

SIS 模型将人群分为 S 类（易感者）和 I 类（感染者），适用于只有易感者和患病者两类人群，可以治愈，但会反复发作的疾病，例如脑炎、细菌性痢疾等治愈后也不具有免疫力的传染病[1]。而新冠作为一种能够反复感染的疾病，用SIS模型来诠释较为恰当。

小世界网络模型是一种复杂网络模型，它模拟了社会网络中的结构，其中的节点之间存在着一定的连接关系。小世界网络模型的特点是，节点之间的平均距离很短，而且节点之间的连接关系也很密集。这种网络模型可以用来模拟社会网络中的结构，以及社会网络中的信息传播。对小世界网络上流行病传播模型的研究表明：疾病在小世界网络中的传播比规则网络中更快且更容易[3]。使用小世界模型来对新冠疫情传染病在SIS模型的传播情况进行仿真具有其现实意义。

## 模型假设

首先，SIS模型不考虑出生率和死亡率以及人口流动，它建立在该地区的总人口数保持不变的基础上；其次，易感者无免疫力，与感染者之间有效接触即会被感染，变为患病者，并且无潜伏期；最后，SIS模型以一天作为模型的最小时间单元，不考虑疾病的变异。

小世界网络模型是一种随机图，它假设社会网络中的节点之间存在着一定的连接关系，而且节点之间的平均距离很短，而且节点之间的连接关系也很密集，也就是平均最短路径小，集聚系数大。

## 模型分析

假设总人数为N，感染率为λ，治愈率为μ，S（t）为当前系统下易感者人数占比，I（t）为当前系统下感染者人数占比，则每日感染的人数为 ，而每日治愈的人数为（图1）。

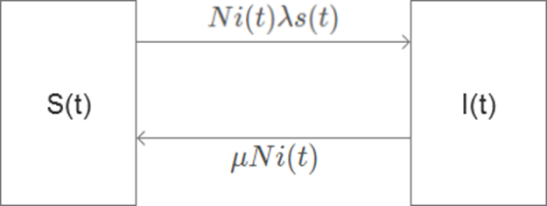


图1 SIS模型图示(图片来源于网络：https://img-blog.csdnimg.cn/20210129222406933.png)

根据上述的转化关系，我们可以得到下列微分方程组[1，2，6]（图2）。

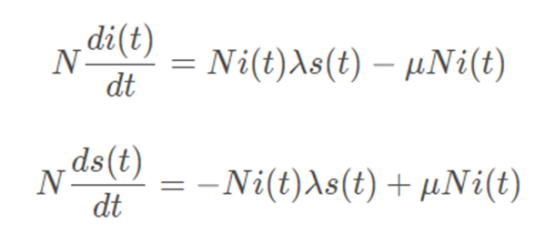
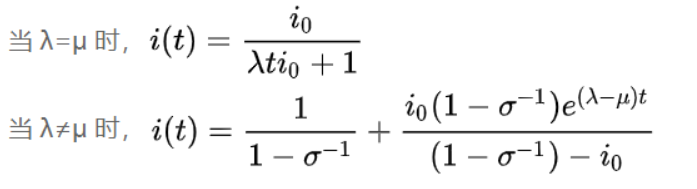
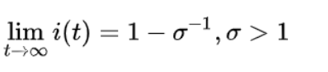


图2 SIS模型微分方程组(图片来源于网络：https://img-blog.csdnimg.cn/20210129222406933.png)

下图（图3）中的传染期接触数为，λ为日接触率，μ为日治愈率，平均治愈天数为，经过化简可知，当σ>1时，患者将会始终存在，并且随着σ（接触数）的增大而趋近于1，而当σ<=1时，患者最终均会痊愈。由此可以得知，σ是一个非常重要的关键变量！同时，通过取极限，我们也能够推断出当疫情稳定时感染人数占比趋向的极限为1-σ-1（图3）。





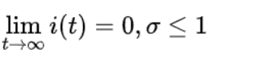


图3 SIS模型结论(图片来源于网络：https://img-blog.csdnimg.cn/20210129222313628.png?x-oss-process=image/watermark,type\_ZmFuZ3poZW5naGVpdGk,shadow\_10,text\_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3UwMTM1OTg5NTc=,size\_16,color\_FFFFFF,t\_70)

## 模型比较

相比于SI模型来说，SIS模型能够较好的反应传染的真实情况，几乎不会出现SI模型中最终所有人都感染的极端情况。而通过分析可知，SI模型是SIS模型μ=0时的特解，也就是日治愈率为0下的特解。

同时，通过比较SIS模型的解析解与数值解（图4），我们也能够发现：SIS模型的数值解与解析解之间的差异较小，模型的数值解的精度高，因此，在SIS模型下，用数值解的方式来拟合、预测新冠疫情是一种行之有效的方法。

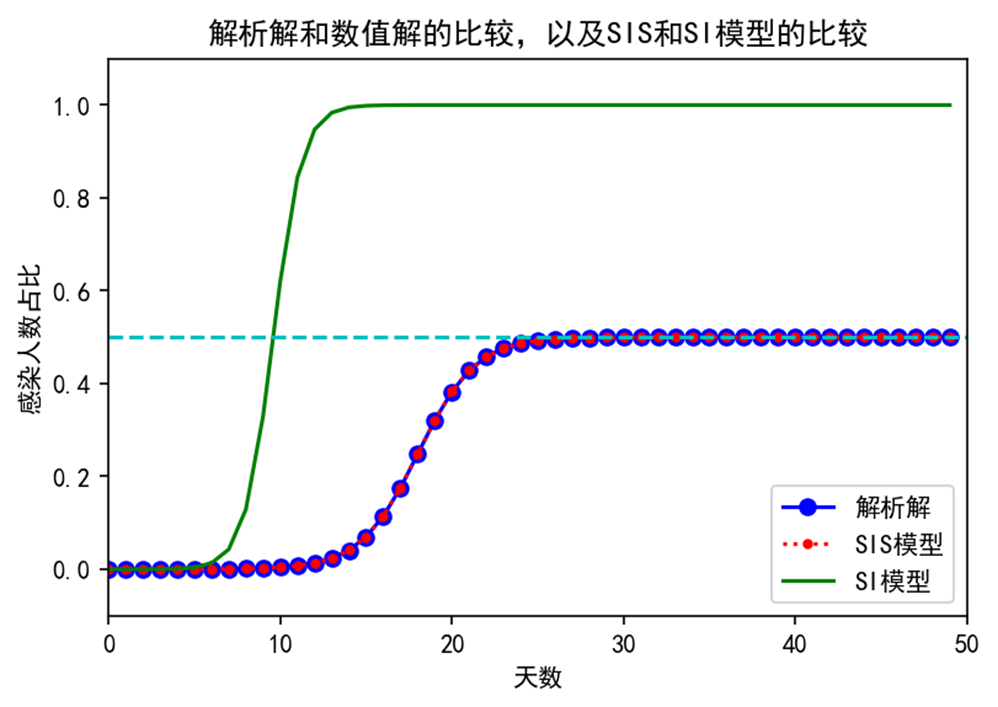


图4 数值解和解析解的比较，以及SIS和SI模型的比较

# SIS模型对于σ参数的分析

从上文中我们可以看到，σ参数对传染病的传播具有重要意义，接下来，本文将对σ参数进行分析。

从下图（图5）中可以看到，当σ大于1的时候，易感者和感染者的比例将会各自收敛到一个数值，这是由于σ>1时产生了地方平衡点，疫情稳定但不会清零，随着时间的增长，患病率将会长期在一个大致稳定的数值上下波动。

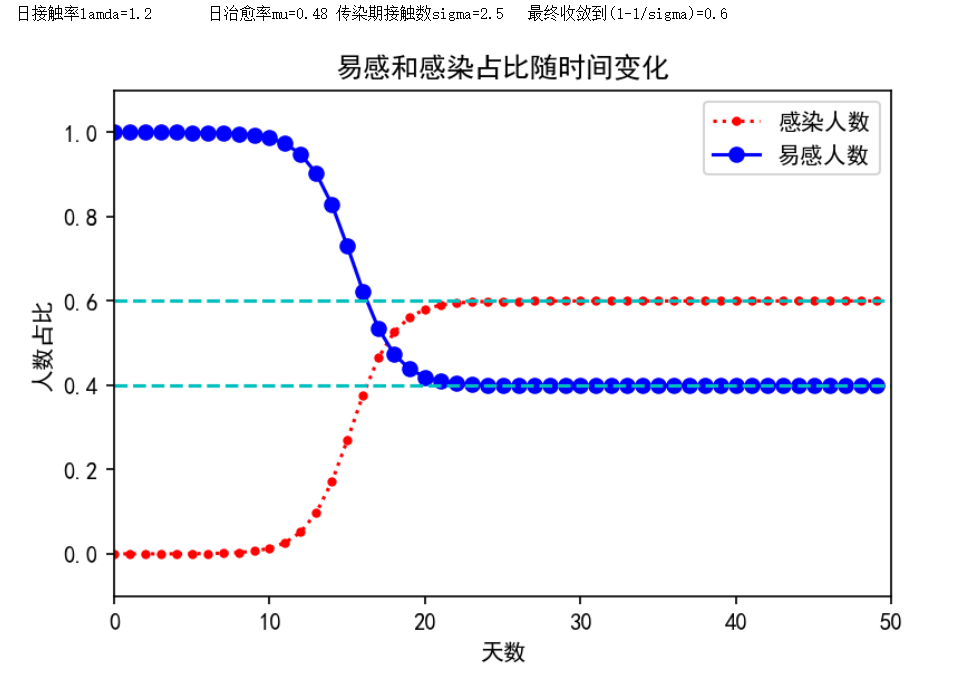


图5σ=2.5时易感和感染占比随时间的变化

直观地看，若每日治愈的人数大于感染的人数，则疫情将会渐渐消失，反之不然。也就是说，当感染期接触数σ大于1时，疫情将会长期存在，成为地方病，若小于1，则疫情将会随时间而逐渐消失。Brusset团队对传染病在SIS模型下的长期变化实验也证明了这一结论[6]。

## 不同σ下的感染人数占比随时间变化

### σ>1

# 当σ>1时（图6），传染期接触率大于1，日接触率大于日治愈率。当患病率小于极限值1-σ-1时，患病率先以一个较快的速率上升，随后逐渐趋于定值，疫情曲线单调上升收敛。而当患病率大于极限值1-σ-1时，患疫情曲线单调下降收敛。结论：当σ>1时疫情终将稳定但不会清零，而是长期保持一定的患病率，称为地方病平衡点。

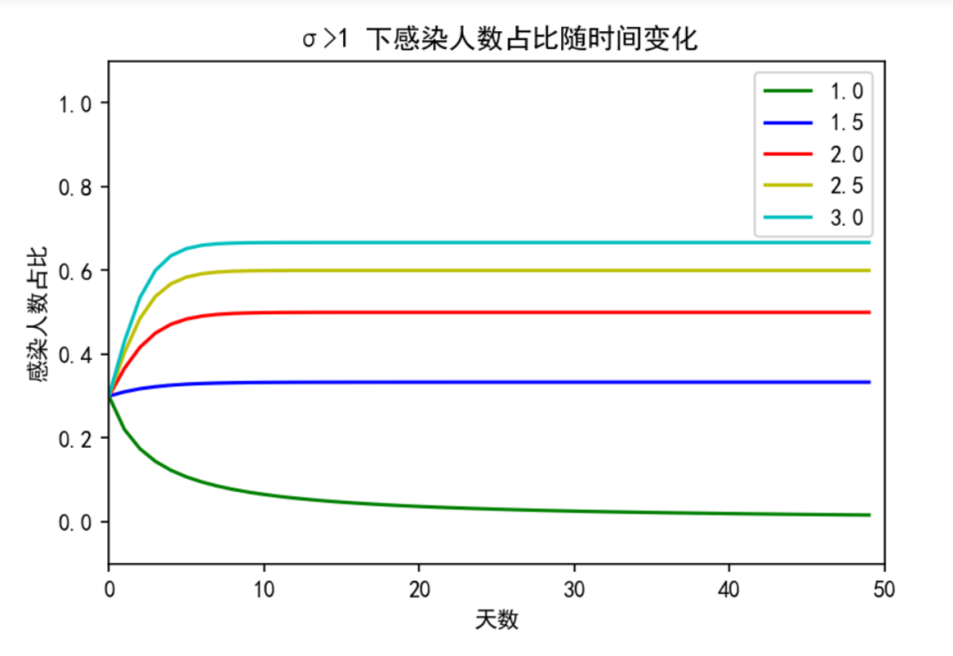


图6 σ>1下感染人数占比随时间变化

### σ=1

当σ=1时（图7），不论患病率初值为多少，患病率仍是单调下降，最终趋近于零。虽然在数学上不可能达到零，但对于现实世界来说，感染者和易感者的人数需要取整，因此疫情最终会清零。

### σ<1

当σ<1时（图7），传染期接触数量小于1，患病率单调下降，最终也会趋近于0，σ越小，疫情越快消失。

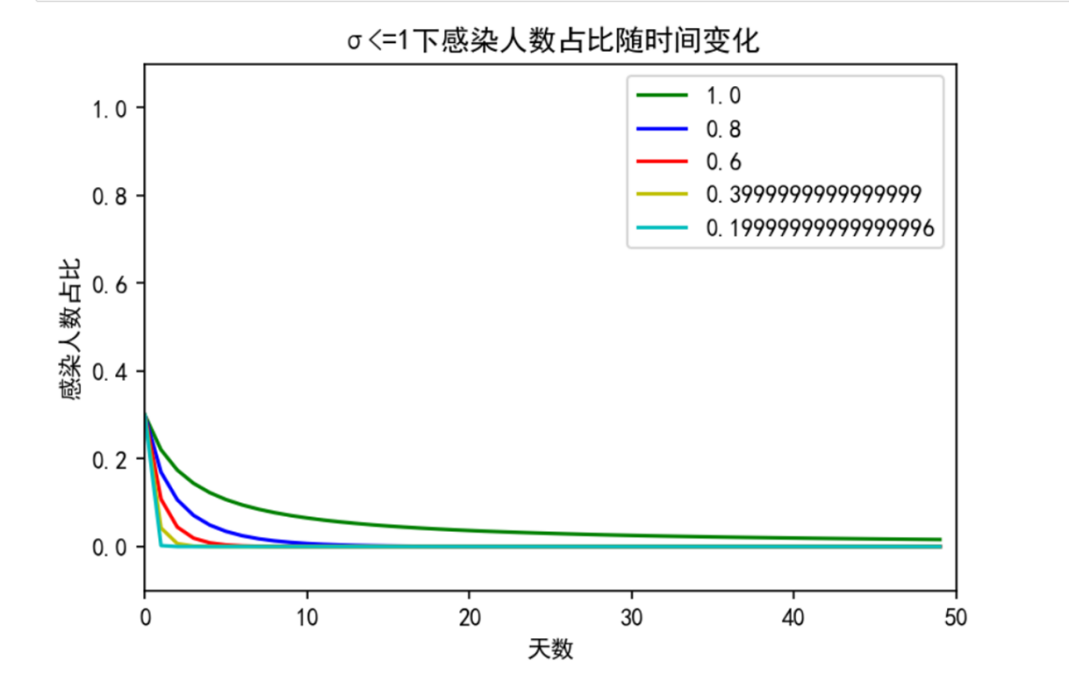


图7 σ<=1下感染人数占比随时间的变化

## 不同σ下时间和di/dt的关系

SIS模型下不同σ和感染人数占比的变化是明显的，但其具体的变化速率却是笑更进一步探讨的，它能够预测疫情增长最快的时间点，帮助我们做出更好地决策。

下图（图8）展示了不同σ下的变化规律，在SIS模型下，当σ>1时，增长速率先单调上升到最高点，随后下降并趋近于0，进一步验证了σ>1时最终疫情将会平稳的结论。而σ越大，增长速率越快，并且峰值越高。而在σ<1时，降低速率单调下降直至疫情消失。

结论：在SIS模型下，传染期接触数越大，疫情爆发初期传播速率越高，越难以控制，减小传染期接触数能够有效地降低传播速率。同时，SIS模型下，疫情的爆发期一般在20天内，随后趋于平稳。

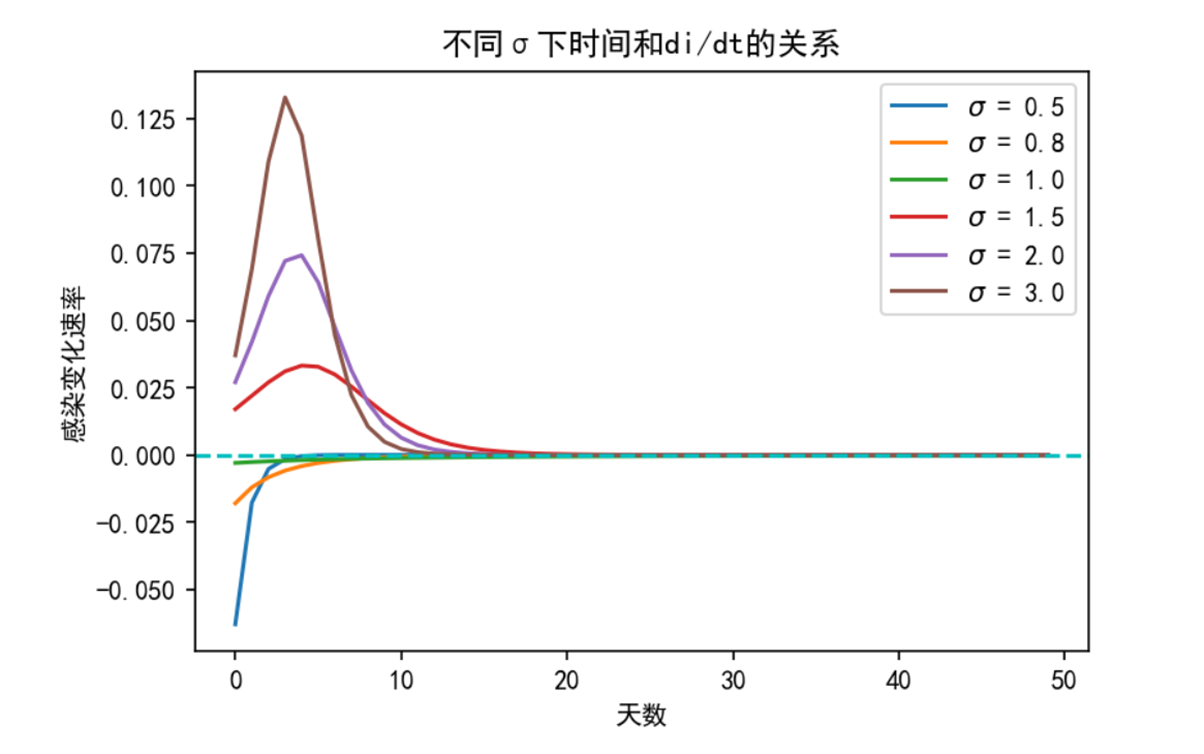


图8 不同σ下时间和di/dt的关系

# SIS模型仿真验证及结论

## NETLOGO软件仿真

为了更好地探讨SIS模型在新冠疫情背景下的真实情况，本实验采用NETLOGO软件进行仿真。设定区域为33\*33，若将每一格看做人走一步的长度（0.6m），区域大小约为392m2，若保持1m的社交距离（人口密度为1人/m2），该区域的人口数约为396人，平均治愈时间1/u为7天，每一个点对应一个人，每一时刻所有的点向任意位置移动一格，初始传染者设置为随机1人。

在1m社交距离的情况下，最终感染人数占比约为57%，σ约为2.32，疫情一旦蔓延，仍易造成大规模的感染。该图像（图9）与SIS模型的解析解的曲线基本一致。

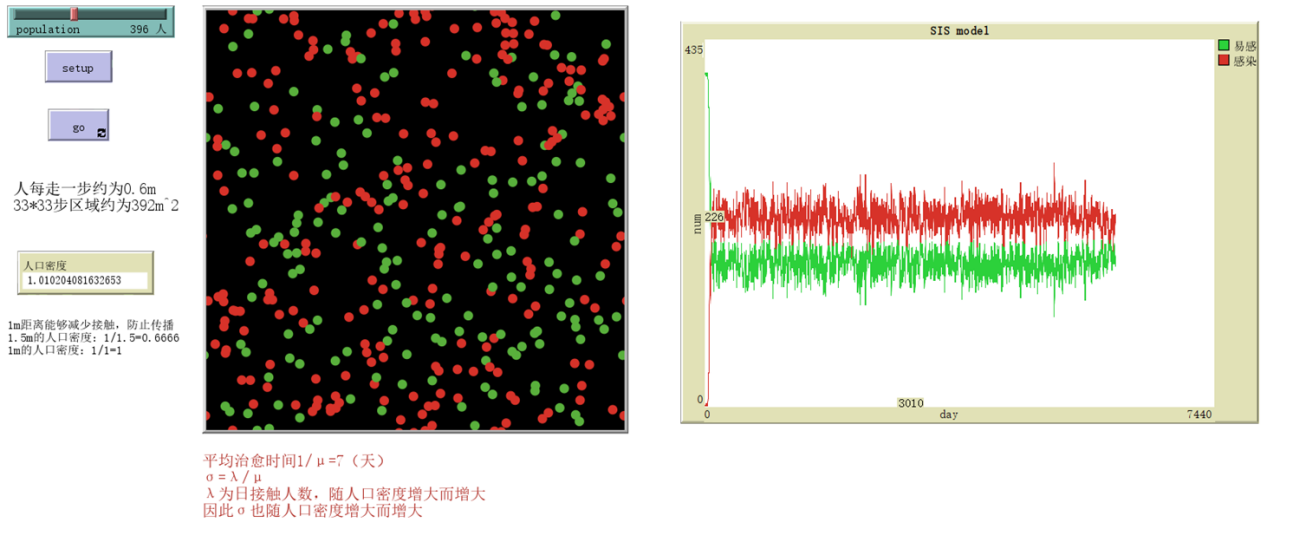


图9 1m社交距离下NETLOGO软件仿真

但有时也会出现下图（图10）的情况，这说明保持1m距离有概率能够让疫情在彻底蔓延之前从根源上解决问题，一定程度上能够防止疫情的蔓延。

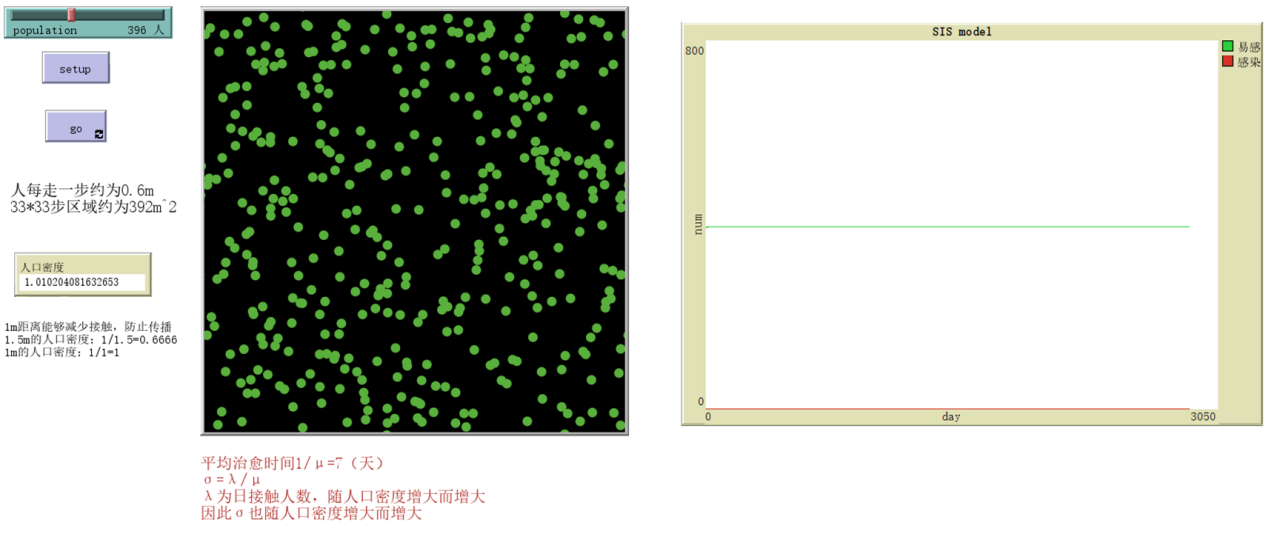


图10 1m社交距离下特殊情况

而在社交距离保持1.5米的情况下，区域内人数为260人，σ减小到1.36，最终感染人数占比约为2.7%（图11）。虽然疫情仍不能消失，但特殊情况出现的概率明显提升，这意味着1.5m的社交距离能够有效地从根源遏制住传染病的传播，达到在传播初期消灭疫情的效果。同时，保持1.5m的社交距离也能够降低最终感染的人数占比，这说明现实生活中提倡的保持1m或1.5m的社交距离有其科学依据。

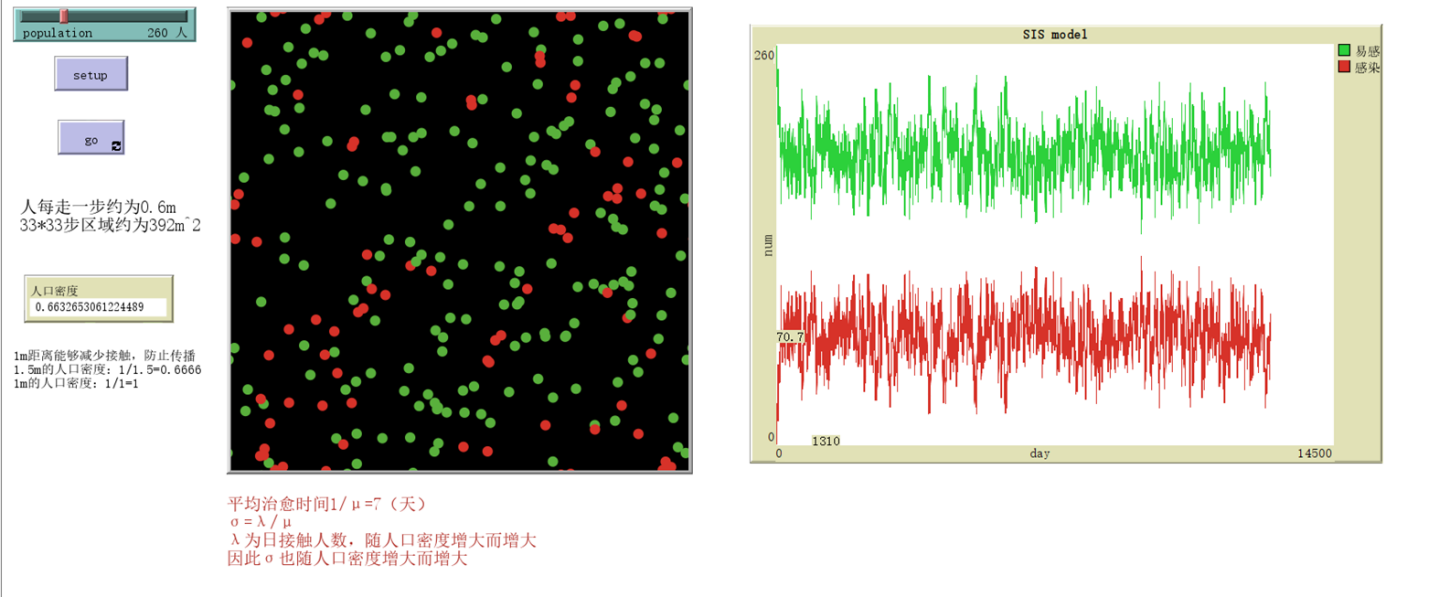


图11 1.5m社交距离下NETLOGO软件仿真

## 上海真实数据SIS模型仿真

12月13日，我国全面放开，疫情迅速蔓延。然而，虽然我们感觉周围的人的感染率很高，但仍没有一个直观的数据，为此，本文截取了上海开放前的疫情数据通过SIS模型进行预测。如图（图12）为上海开放前疫情的部分数据（数据来源：上海市卫生健康委员会），可以发现，在12月10-13日，上海的平均现存病例为298人，康复率为1/7，新增确诊约154人，平均日接触率为0.52，σ为3.64。



图12 上海开放前疫情部分数据（数据来源：上海市卫生健康委员会）

若以这种情况放开，通过SIS模型进行预测，疫情将会在10天（12月23日）后进入爆发期，在15天（12月27日）左右增长速度达到最大，随后逐步放缓，在25天（1月7日）左右进入平稳期，最终将会有73%以上的人被感染。

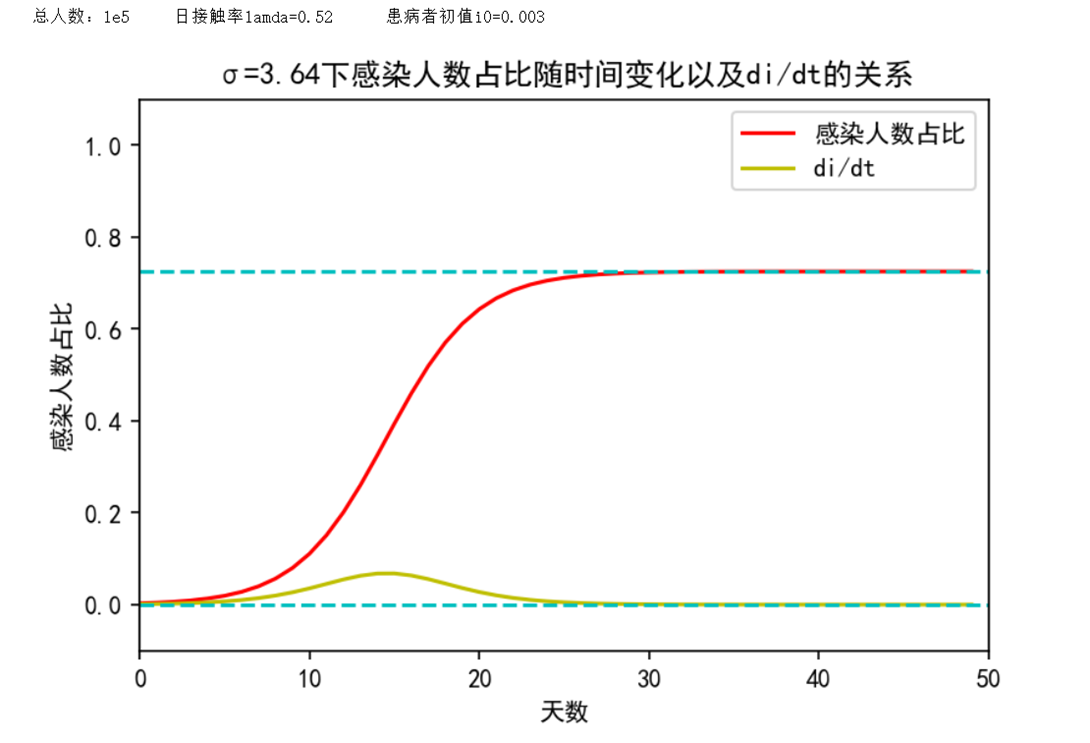


图13 SIS模型下上海开放后疫情预测

## 上海真实数据小世界网络仿真

小世界网络最初是由美国物理学家和数学家弗雷德里克·沃尔夫（Frederick Watts）和罗伯特·斯特恩（Robert Ston）在1998年提出的。小世界网络具有短平均路径以及高集聚系数的特点，能够有效地模拟社会网络，模拟传染病在社会网络下的传播规律[3，5]。

为了更好地了解新冠在人群中的传播，本实验通过使用小世界网络这一复杂网络进行仿真模拟。

本次小世界网络设置了298个节点，每个节点6个邻居，随机化重连的概率为0.5（图14）。

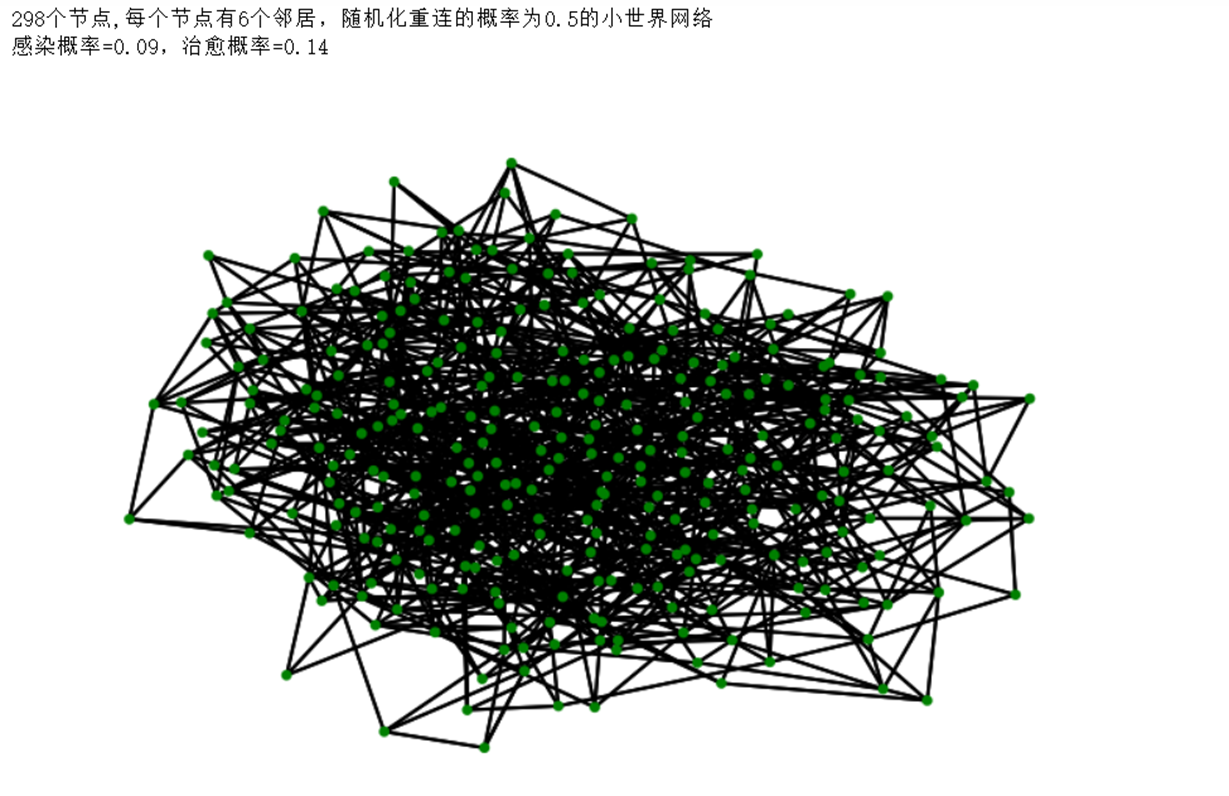
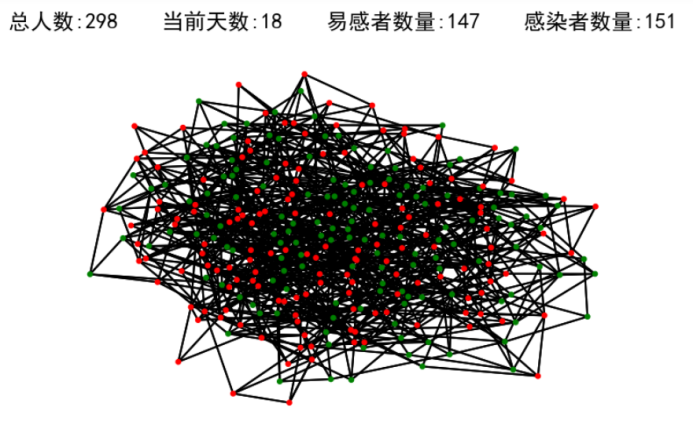
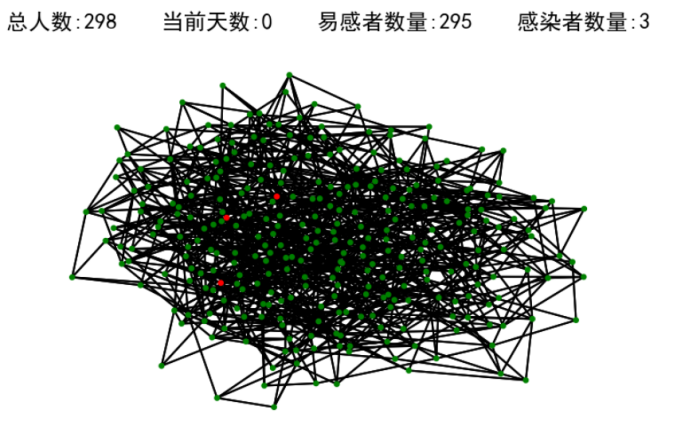


图14 小世界网络构建

本次实验挑选了中心度最高的三个点作为感染源，进行传播，传播时间长度设置为50天，每个节点的感染概率为0.09，治愈概率为0.14。可以看到（图15、16），传染病迅速在整个网络中传播开来，感染者比例逐渐趋于稳定。



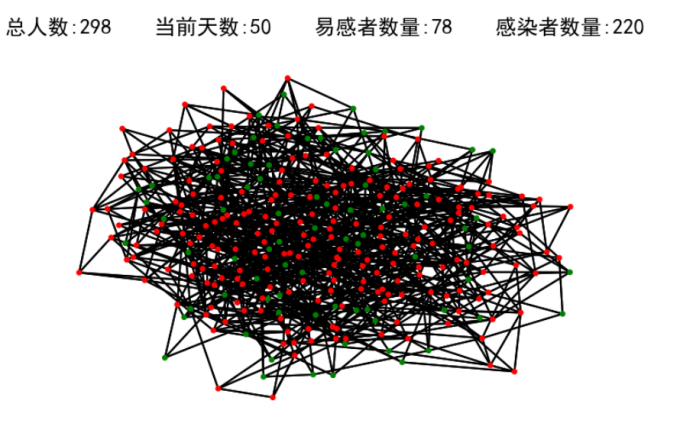
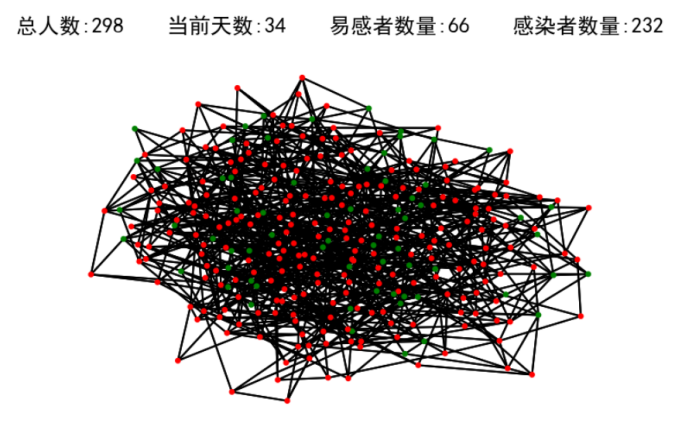


图15 SIS模型小世界网络疫情传播仿真图像

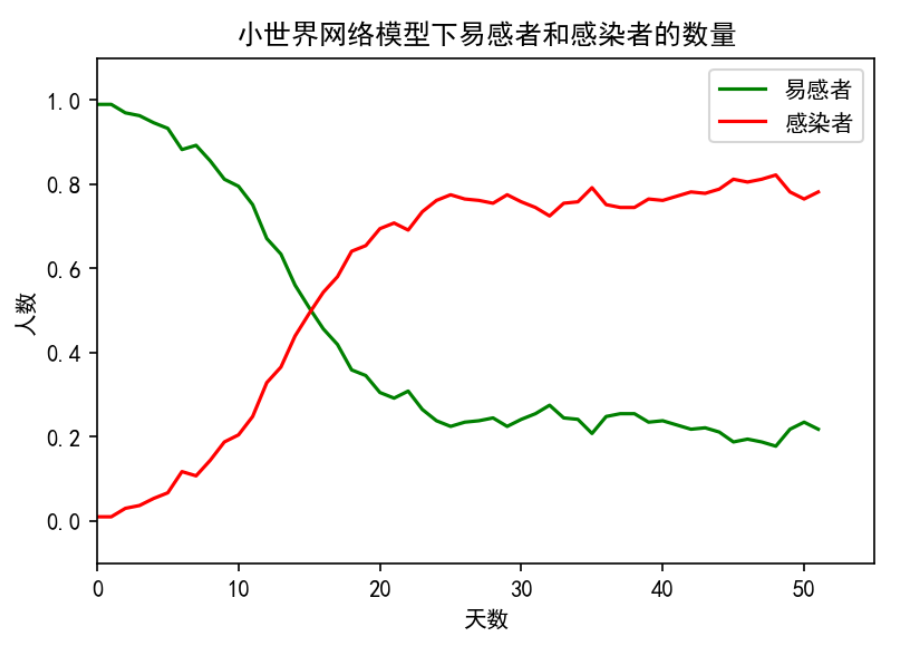


图16 SIS模型小世界网络仿真易感者和感染者占比折线图

## 小世界网络与解析解仿真对比

下图（图17）中可以看到，若调整小世界网络节点的感染率来去拟合SIS模型的解析解，本文通过多次试验发现，当小世界模型的感染率设置为0.09时，拟合的程度较好。也就是说，若疫情如SIS模型预测的那样增长，则在我国全面放开后，当感染者与易感者之间进行接触时，易感者将会有9%的概率被感染。

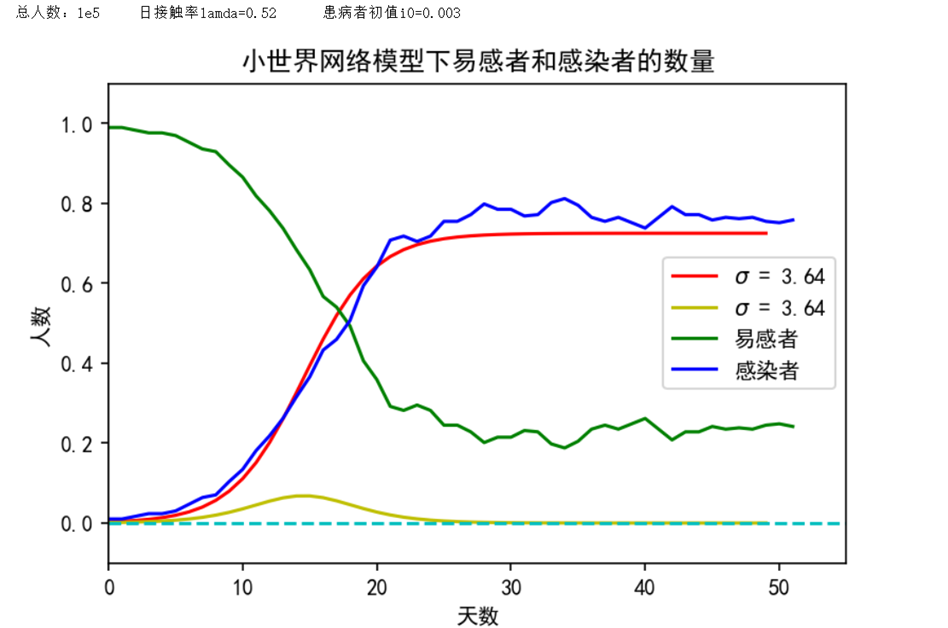


图17 SIS模型小世界网络和解析解易感者和感染者占比折线图

# SIS模型的缺陷

对于流感等反复感染的病毒，SIS模型能够进行大致的刻画，帮助我们把握关键（σ），从而更好地打破病毒传播链。通过SIS模型可以预测一个封闭地区疫情的爆发情况、最大峰值、感染人数等等，但这一切都建立在封闭的条件下。而外部因素，如相对距离、外部区域人口、外部区域人口流动等，SIS模型都难以考虑，也无法解决超级传播者（在病毒传播初期，超级传播者的影响非常大）和潜伏期问题（病人在潜伏期内，也可能具有传染性）等问题。

参考文献

1. 徐涵,张庆.复杂网络上传播动力学模型研究综述[J].情报科学, 2020,38(10):159-167.D OI:10.13833/j.issn.1007-7634.2020.10.024.

[2]孙敏淇. 复杂网络上包含相关系数的SIS动力学模型建立与分析[D].山西大学, 2021. DOI:10.27284/d.cnki.gsxiu.2021.001176.

[3]韩克春,张晓勇,吴海英等.基于复杂网络的流行病传播SIS模型仿真研究[J].现代预防医学,2014,41(20):3649-3651.

[4] Leonid E. Zhukov.Epidemics on networks II[R].National Research University Higher School of Economics ,2020

[5]王佳亮,李海滨,李海燕.基于复杂网络的新冠病毒群体免疫数值仿真[J/OL].复杂系统与复杂性科学:1-8[2023-02-23].http://kns.cnki.net/kcms/detail/37.1402.N.20221021.1652.014.html.

[6]Brusset, X., Davari, M., Kinra, A., Torre, D.L. (2021). Modelling COVID-19 Ripple Effect and Global Supply Chain Productivity Impacts Using a Reaction-Diffusion Time-Space SIS Model. In: Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., Romero, D. (eds) Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems. APMS 2021. IFIP Advances in Information and Communication Technology, vol 633. Springer, Cham. https://doi.org/10.1007/978-3-030-85910-7\_1

### 核心代码附录

|  |
| --- |
| # 解析解和数值解的比较，以及SIS和SI模型的比较  # 设置模型参数  number = 1e5 # 总人数  lamda = 1.2 # 日接触率, 患病者每天有效接触的易感者的平均人数  sigma = 2 # 传染期接触数  mu = lamda/sigma # 日治愈率, 每天被治愈的患病者人数占患病者总数的比例  fsig = 1-1/sigma #t->无穷 时最终收敛到的值  y0 = i0 = 1e-5 # 患病者比例的初值  tEnd = 50 # 预测日期长度  t = np.arange(0.0,tEnd,1) # (start,stop,step)  print("日接触率lamda={}\t日治愈率mu={}\t传染期接触数sigma={}\t最终收敛到(1-1/sigma)={}".format(lamda,mu,sigma,fsig))  # 解析解  if lamda == mu:  Analy = 1.0/(lamda\*t +1.0/i0)  else:  Analy= 1.0/((lamda/(lamda-mu)) + ((1/i0)-(lamda/(lamda-mu))) \* np.exp(-(lamda-mu)\*t))  # odeint 数值解，求解微分方程初值问题  SI = odeint(dy\_dt, y0, t, args=(lamda,0)) # SI 模型  SIS = odeint(dy\_dt, y0, t, args=(lamda,mu)) # SIS 模型  # 绘图  plt.plot(t, Analy, '-ob', label='解析解')  plt.plot(t, SIS, ':.r', label='SIS模型')  plt.plot(t, SI, '-g', label='SI模型')  plt.title("解析解和数值解的比较，以及SIS和SI模型的比较")  plt.axhline(y=fsig,ls="--",c='c') # 添加水平直线  plt.legend(loc='best') # youcans  #设置x轴的范围为[a, b]，y轴的范围为[c, d]  plt.axis([0, 50, -0.1, 1.1])  plt.xlabel("天数")  plt.ylabel("感染人数占比")  plt.show()  # 不同σ下感染人数占比随时间变化  # 设置模型参数  number = 1e5 # 总人数  lamda = 1.2 # 日接触率, 患病者每天有效接触的易感者的平均人数  y0 = i0 = 0.3 # 患病者比例的初值  tEnd = 50 # 预测日期长度  t = np.arange(0.0,tEnd,1) # (start,stop,step)  print("总人数：1e5\t日接触率lamda={}\t患病者初值i0=0.3".format(lamda,mu,sigma,fsig))  drowNum=5  lineColor=['g','b','r','y','c','m']  for i in range(0,drowNum):  sigma=1+0.5\*i  mu = lamda/sigma # 日治愈率, 每天被治愈的患病者人数占患病者总数的比例  infect = odeint(dy\_dt, y0, t, args=(lamda,mu)) # 感染人数  plt.plot(t, infect,lineColor[i], label=sigma)  plt.title("σ>1 下感染人数占比随时间变化")  plt.legend(loc='best') # youcans  plt.axis([0, 50, -0.1, 1.1])  plt.xlabel("天数")  plt.ylabel("感染人数占比")  plt.show()  sigma = np.array((0.5, 0.8, 1.0, 1.5, 2.0, 3.0))  for i in range(0,drowNum):  sigma=1-0.2\*i  mu = lamda/sigma # 日治愈率, 每天被治愈的患病者人数占患病者总数的比例  infect = odeint(dy\_dt, y0, t, args=(lamda,mu)) # 感染人数  plt.plot(t, infect,lineColor[i], label=sigma)  plt.title("σ<=1下感染人数占比随时间变化")  plt.legend(loc='best') # youcans  plt.axis([0, 50, -0.1, 1.1])  plt.xlabel("天数")  plt.ylabel("感染人数占比")  plt.show()  # 不同σ下时间和di/dt的关系  # 设置模型参数  number = 1e5 # 总人数  lamda = 1.2 # 日接触率, 患病者每天有效接触的易感者的平均人数  sigma = np.array((0.5, 0.8, 1.0, 1.5, 2.0, 3.0)) # 传染期接触数  y0 = i0 = 0.05 # 患病者比例的初值  tEnd = 50 # 预测日期长度  t = np.arange(0.0,tEnd,1) # (start,stop,step)  for x in sigma:  SIS = odeint(dy\_dt, y0, t, args=(lamda,lamda/x))  Deriv = lamda\*SIS\*(1-SIS) - SIS\*lamda/x  plt.plot(t, Deriv, '-', label=r"$\sigma$ = {}".format(x)) #label='di/dt~i'  print("lamda={}\tmu={}\tsigma={}\t(1-1/sig)={}".format(lamda,lamda/x,x,(1-1/x)))  # 绘图  plt.axhline(y=0,ls="--",c='c') # 添加水平直线  plt.title("不同σ下时间和di/dt的关系")  plt.legend(loc='best')  plt.xlabel("天数")  plt.ylabel("感染变化速率")  plt.show() **# 真实数据数学方程下的图像** # 设置模型参数  lamda = 0.52 # 日接触率, 患病者每天有效接触的易感者的平均人数  y0 = i0 = 0.003 # 患病者比例的初值  tEnd = 50 # 预测日期长度  t = np.arange(0.0,tEnd,1) # (start,stop,step)  print("总人数：1e5\t日接触率lamda={}\t患病者初值i0=0.003".format(lamda,0.14,3.64,0.73))  sigma=3.64  fsig=1-1/sigma  mu = 0.14 # 日治愈率, 每天被治愈的患病者人数占患病者总数的比例  infect = odeint(dy\_dt, y0, t, args=(lamda,lamda/sigma)) # 感染人数  plt.plot(t, infect,"r", label=r"$\sigma$ = 3.64")  plt.title("σ=3.64下感染人数占比随时间变化以及di/dt的关系")  plt.legend(loc='best') # youcans  plt.axis([0, 50, -0.1, 1.1])  plt.xlabel("天数")  plt.ylabel("感染人数占比")  plt.axhline(y=fsig,ls="--",c='c') # 添加水平直线  #速率  Deriv = lamda\*infect\*(1-infect) - infect\*lamda/sigma  plt.plot(t, Deriv, 'y', label=r"$\sigma$ = {}".format(sigma)) #label='di/dt~i'  plt.axhline(y=0,ls="--",c='c') # 添加水平直线  plt.legend(loc='best')  plt.xlabel("天数")  plt.show()  #小世界网络仿真  #来生成一个有N个节点,每个节点有4个邻居，随机化重连的概率为p的随机网络  print("298个节点,每个节点有6个邻居，随机化重连的概率为0.5的小世界网络")  print("感染概率=0.09，治愈概率=0.14")  N = 298  p = 0.5  ws=nx.watts\_strogatz\_graph(N,6,p)  for i in range(N):  ws.nodes[i]['state'] = 'S'  gamma = 1/7  beta = 0.09  ps=nx.spring\_layout(ws)#布置框架  colors={"I":'r',"S":'g'}  states= nx.get\_node\_attributes(ws, 'state')############ 获得节点的isCore属性  color=[colors[states[i]] for i in range(N)]  nx.draw\_networkx\_edges(ws,ps,width=1)  nx.draw(ws,ps,node\_color =color ,node\_size=5)  plt.show()  tEnd = 50 # 预测日期长度  def spread(G,beta,initial,func,gamma = 0 ):  colors={"I":'r',"S":'g'}  y = []  n = len(G.nodes)#总人数  for i in range(n):#所有人默认为易感染  G.nodes[i]['state'] = 'S'  s = n - initial #易感染人数  desc\_dc = func(G)    i\_nodes = []  #选择前inttial个度中心性最高的节点设为感染源  for i in range(initial):  G.nodes[desc\_dc[0][0]]['state'] = 'I'  i\_nodes.append(desc\_dc[0][0])  desc\_dc.remove(desc\_dc[0])  y.append((s,len(i\_nodes)))  #开始传播，传播50天tEnd  nowday=0;  #初始图像  print("总人数:",n)  print("当前天数",nowday)  print("易感者数量：",s)  print("感染者数量",n-s)  s = n - len(i\_nodes)  i = len(i\_nodes)  y.append((s,i))  states= nx.get\_node\_attributes(G, 'state')############ 获得节点的属性  color=[colors[states[i]] for i in range(n)]  nx.draw\_networkx\_edges(ws,ps,width=1)  nx.draw(ws,ps,node\_color =color ,node\_size=5)  phototitle="总人数:"+str(n)+" 当前天数:"+str(nowday)+" 易感者数量:"+str(s)+" 感染者数量:"+str(n-s)  plt.title(phototitle)  plt.show()  while nowday!=tEnd :  nowday=nowday+1  #当前轮被传染的人数  i\_temp = []  #当前恢复人数 gamma 概率  for i in i\_nodes:  if random.random() < gamma:  i\_nodes.remove(i)  G.nodes[i]['state'] = 'S'  i\_nodes\_temp = nx.Graph()  i\_nodes\_temp.add\_nodes\_from(i\_nodes)  for i in i\_nodes\_temp.nodes:  #按beta概率传染I节点的邻居节点  for node in G.neighbors(i):  r= random.random()  if r < beta and G.nodes[node]['state'] == 'S':  G.nodes[node]['state'] = 'I'  i\_temp.append(node)  for t in i\_temp :  if t not in i\_nodes:  i\_nodes.append(t)  s = n - len(i\_nodes)  i = len(i\_nodes)  y.append((s,i))  states= nx.get\_node\_attributes(G, 'state')############ 获得节点的属性  color=[colors[states[i]] for i in range(n)]  nx.draw\_networkx\_edges(ws,ps,width=1)  nx.draw(ws,ps,node\_color =color ,node\_size=5)  print("总人数:",n)  print("当前天数",nowday)  print("易感者数量：",s)  print("感染者数量",n-s)  phototitle="总人数:"+str(n)+" 当前天数:"+str(nowday)+" 易感者数量:"+str(s)+" 感染者数量:"+str(n-s)  plt.title(phototitle)  fig=plt.gcf()  plt.show()  strday="./"+str(nowday)+".jpg"  fig.savefig(strday)  return np.array(y,dtype=object)  #选择度中心性最高的3个点作为感染源  result = spread(ws,beta,3,centrality,gamma)  print(result)  plt.title("小世界网络模型下易感者和感染者的数量")  suspect=result[:,0]  suspect=[x/N for x in suspect ]  infect=result[:,1]  infect=[x/N for x in infect ]  print("298个节点,每个节点有6个邻居，随机化重连的概率为0.5的小世界网络")  print("感染概率=0.09，治愈概率=0.14")  plt.plot(suspect, 'g', label='易感者')  plt.plot(infect, 'r', label='感染者')  plt.legend(loc='best') # youcans  #设置x轴的范围为[a, b]，y轴的范围为[c, d]  plt.axis([0, tEnd+5, -0.1, 1.1])  plt.xlabel("天数")  plt.ylabel("人数")  plt.show()  # 设置模型参数  lamda = 0.52 # 日接触率, 患病者每天有效接触的易感者的平均人数  y0 = i0 = 0.003 # 患病者比例的初值  tEnd = 50 # 预测日期长度  t = np.arange(0.0,tEnd,1) # (start,stop,step)  print("总人数：1e5\t日接触率lamda={}\t患病者初值i0=0.003".format(lamda,0.14,3.64,0.73))  print("298个节点,每个节点有6个邻居，随机化重连的概率为0.5的小世界网络")  print("感染概率=0.09，治愈概率=0.14")  sigma=3.64  fsig=1-1/sigma  mu = 0.14 # 日治愈率, 每天被治愈的患病者人数占患病者总数的比例  infect = odeint(dy\_dt, y0, t, args=(lamda,lamda/sigma)) # 感染人数  plt.plot(t, infect,"r", label=r"$\sigma$ = 3.64")  #速率  Deriv = lamda\*infect\*(1-infect) - infect\*lamda/sigma  plt.plot(t, Deriv, 'y', label=r"$\sigma$ = {}".format(sigma)) #label='di/dt~i'  plt.axhline(y=0,ls="--",c='c') # 添加水平直线  plt.title("小世界网络模型下易感者和感染者的数量")  suspect=result[:,0]  suspect=[x/N for x in suspect ]  infect=result[:,1]  infect=[x/N for x in infect ]  plt.plot(suspect, 'g', label='易感者')  plt.plot(infect, 'b', label='感染者')  plt.legend(loc='best') # youcans  plt.axis([0, tEnd+5, -0.1, 1.1])  plt.xlabel("天数")  plt.ylabel("人数")  plt.show() |