廣東工業大學

QG 中期考核详细报告书

题　　目	多智能体协同系统
学　　院	计算机学院
专　　业	计算机类
年级班别	2023 级 2 班
学　　号	3123004200
学生姓名	许国伟

2024 年 4 月 5 日

目录

工作概述

1. 详细阅读论文<<多智能体聚集问题研究综述>>by 李杨,大致了解多智能体系统的含义,了解什么是多智能体聚集

2. 先略读<<On Krause’s Multi-Agent Consensus Model>>,了解多智能体的大致框架和基本知识

3. 再详解<<On Krause’s Multi-Agent Consensus Model>>,理解其中涉及的数学公式,手动推导系统用到的数学结论并做相关记录

4. 重点研究<<On Krause’s Multi-Agent Consensus Model>>中的实验,借助论文中关系图剖析实验原理并思考实验意义

5. 对照<<On Krause’s Multi-Agent Consensus Model>>中的关系图和数学公式利用 python 代码复现实验

6. 记录学习过程,写下详细文档,制作答辩 ppt

论文阅读

A. 多智能体聚集问题研究综述_李杨

1. 智能体是在某一环境下,能持续自主地发挥作用的计算实体

2. 多智能体系统是由多个智能体在满足某种条件时,能够在环境中交互组成的计算系统

3. 多智能体聚集是指智能体因不断相互影响而靠拢的现象

4. 领导者-跟随者系统是一种多智能体系统.基于互联拓扑的各种模型,领导者在拓扑模型中起到中枢的作用,各个跟随者可以通过领导者中转,进而起到信息交流的目的

5. 多智能体聚集协议指系统内每个智能体的一中属性以及智能体输入的方式

6. 对多智能体聚集协议的优化:

   (1) 周期性更改控制资源的策略 --> 利用事件触发更改

      ① 双事件触发

      ② 判断本地采样信息

      ③ 利用误差进行自触发

      ④ 收集邻居的信息替代误差

   (2) 通过约束输入和切换拓扑进行收敛速度优化

      ① 拓扑建模,将节点之间的边和时变权重相关联

      ② 以领导者为根有向生成树，半全局实现聚集

      ③ 约束异步双坐标下降

      ④ 因子循环和分层循环

   (3) 其他聚集协议优化

      ① 难以预测聚集点 --> 在一定区域内选择聚集点(并利用偏差循环追踪进行了扩大)

      ② 基于时变控制律控制质心聚集

      ③ 鲁棒容错聚集算法处理故障机器人聚集

      ④ 基于图论和非负矩阵理论研究用于任何维度空间的通用议设计框架

⑤ 基于差分博弈论确定在目标处是否聚集取决于领导者对的渴望度。

7. 切换拓扑是一种网络设备连接的方式,它是动态的

8. 固定拓扑同样是一种网络设备连接的方式,它是静态的

B. On Krause’s Multi-Agent Consensus Model

I. INTRODUCTION

论文介绍的智能体模型是周期性地更新位置,更新公式如下(离散模型):

$$x_i(t+1) = \frac{\sum\limits_{j:|x_i(t)-x_j(t)|<1} x_j(t)}{\sum\limits_{j:|x_i(t)-x_j(t)|<1} 1}. \qquad (1)$$

1. 在智能体的不断更新中,我们将每次更新称为世代的迭代,即智能体的初始值为第一世代。

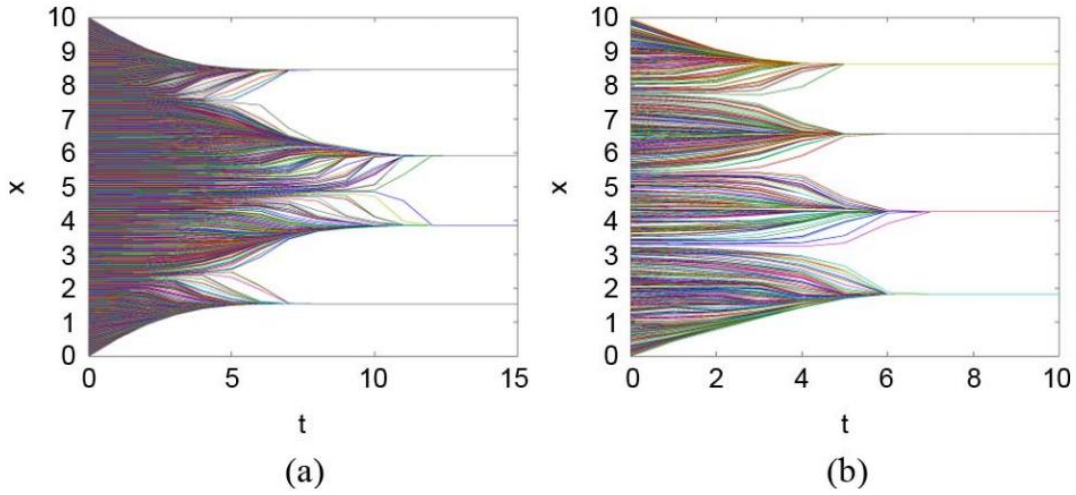
2. 在数学公式(1)中,t 表示某个世代,xi(t)则表示某个世代智能体 i 的位置。

3. 由 j 的解释可知,j 是与 xi 距离小于 1 的智能体集合,在迭代中 xi(t+1)的值由 xj(t)和 xi(t)可得,表示 j 集合里所有元素的平均值。因此对于同一世代的智能体来说,他们位置值的计算是相互独立的

4. 在论文中,我们设距离小于 1 的智能体互为邻居,他们之间存在因果关系

5. 对于智能体的初始值,可随机分配,也可以在某一区间内平均分布,依据使用者的意愿而定

6. 由公式(1)可知,互为邻居的智能体无论经过多少次迭代,仍然互为邻居



1. a 图为拥有区间内均匀分布的初始值的智能体的世代迭代,b 图为拥有随机分布的初始值的智能体的世代迭代。

2. 由图可知,无论是均匀分布还是随机分布,智能体最后聚集的点(以下称为集群)之间的距离总是大于 1 接近 2 的

3. 这是因为如果距离小于 1,那么他们之间会产生因果关系,在下一次迭代中相互影响,进而不能达到稳定的分布

II. THE DISCRETE-AGENT MODEL

A. Basic Properties and Convergence

1. 智能体位置的大小关系取决于初始值的大小关系,即在迭代中位置的大小关系不变,数学表达式如下:

$$\text{if } x_i(0) \leq x_j(0), \text{ then } x_i(t) \leq x_j(t) \text{ for all } t.$$

2. 初始值最小的智能体位置不会再变小,初始值最大的智能体位置不会再变大。即在迭代中,对任意世代 t,都有 x1(t) > x1(0) 和 xn(t) < xn(0),其中 n 为最后一个智能体

3. 在世代的迭代中,如果没有新的智能体引入,整个智能体系统的平均值应是不变的

4. 由公式(1)可知,系统平衡后,对于每个智能体,他们的位置不是等于某个

集群的值,就是与这个集群的距离大于 1

5. 智能体在属于一个集群后就不会再与集群范围外(距离大于 1)的智能体交流

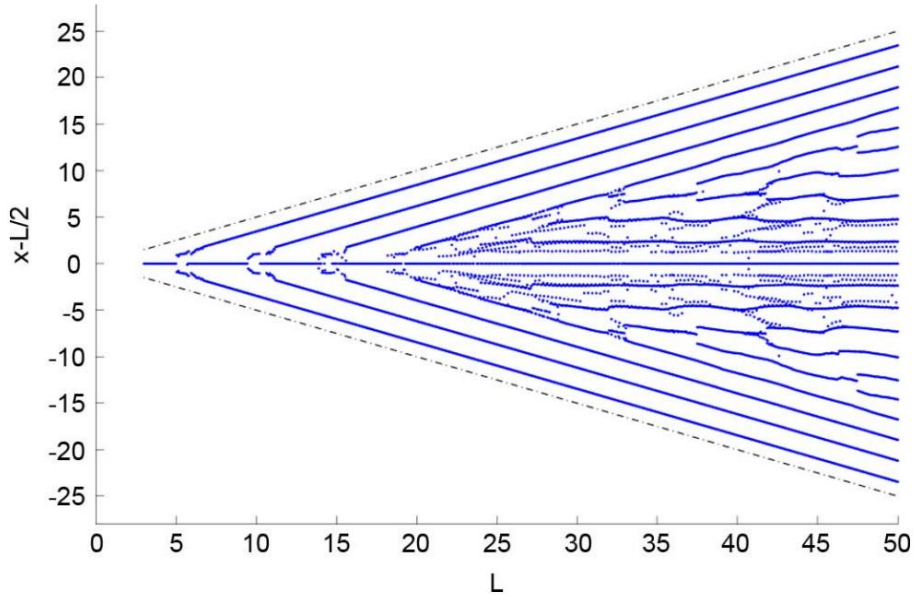
6. 聚集时间与智能体的数量正相关

7. 有限可数的智能体无法满足离散智能体模型的相关条件,而无限可数的智能体可以

B. Experimental Observations

非权重模型

1. 设有 5000L 个智能体均匀分布在[0,L]区间上,让他们依据公式(1)迭代。以 L 为横坐标,x-2/L 为纵坐标,画出对应每个 L 智能体的聚集点有哪些,如图:



2. 论文中展示了在(0,25)上各个 L 对应的聚集点情况,由图可知,聚集点位置为 L/2 几乎在每个 L 都会存在,再联想公式(1)可得:对于智能体总数为奇数的系统,中间值在迭代中是始终不变的,中间值附近的智能体会在迭代中趋向中间偏小或中间偏大。

3. 实践可知,如果智能体随机分布,最后聚集点的情况仍然满足此图

4. 看起来,聚集点与 L 存在一定的线性关系

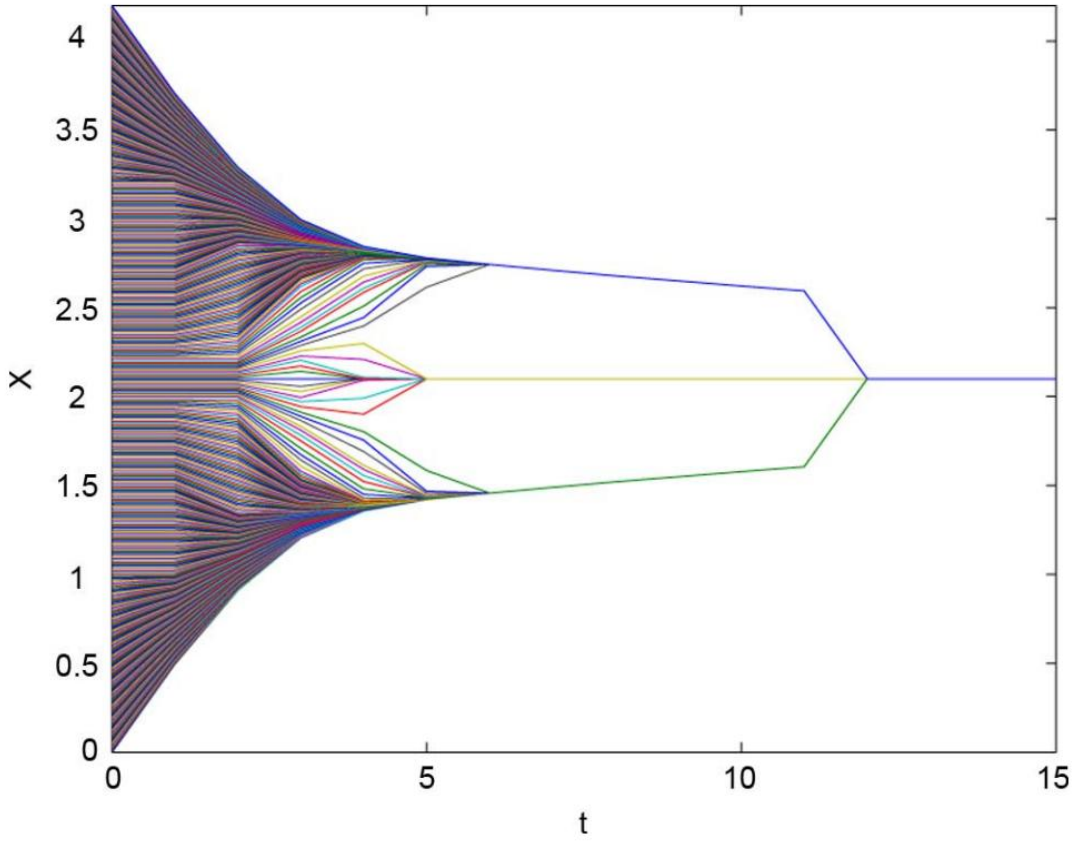
5. 模型仍然存在一些不规则的现象(如每过一段 L,0 轴就会有聚集点的缺

失),这可能是因为特定的离散度,或系统误差的累积

6. 由图可知,L 在大于 3.833 时才会有明显的集群

7. 集群的存在使每个智能体与集群内的智能体互联,与集群外的智能体断连

8. 由图可知,集群间的距离接近 2.2



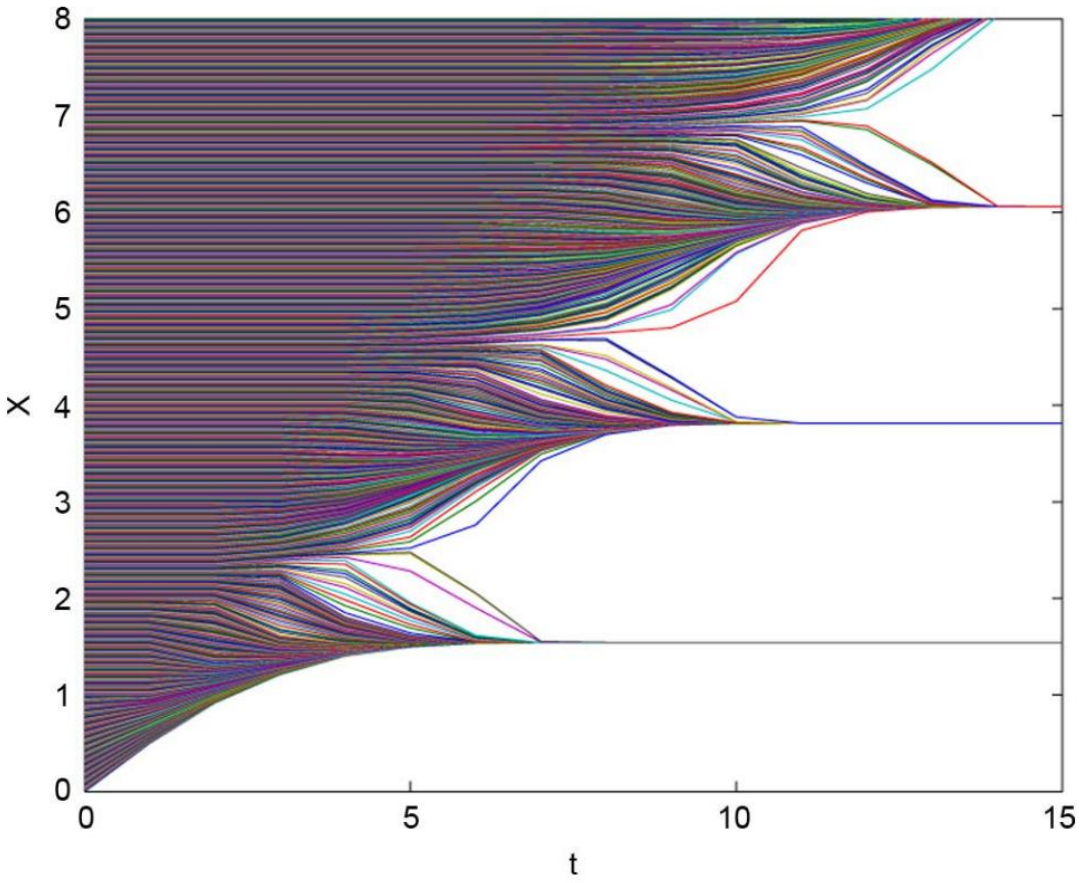
1. 设 L 小于两个集群应有的距离(约为 2.2*2=4.4),在[0,L]上有均匀分布的智能体,以世代 t 为横轴,各个智能体的位置为纵轴,画出他们迭代的过程图,如上图:

2. 由图可知,智能体先形成了两个集群,且集群的距离大于 1.但因为 L<4.4 且集群中间有智能体让两个集群间接交流,所以两个集群必定会相互影响,最终趋于一致,演变成同一个集群

3. 由 2 可推出,如果两个集群间没有智能体使两个集群间接联系,那么两个集群将不会演变为同一个集群,同时得到后文权重模型的理论:中间智能体的权

重可以是任意小的

4. 如果 L>4.4,那么两个集群将不会汇聚,如图:



与上图不同的是,L 增长到了 8,远大于最大距离 4.4。且在这种情况下,即使中间有智能体,也无法让两个集群互联,因为中间智能体不能同时与两个集群联系

权重模型

在这个模型中,我们引入权重,每个智能体除了拥有初始值外,还拥有自己的权重,那么公式(2)会有微小的变化,如下:

$$x_i(t+1)=\frac{\sum_{j:|x_i(t)-x_j(t)|<1} w_j x_j(t)}{\sum_{j:|x_i(t)-x_j(t)|<1} w_j}. \qquad (2)$$

1. 如果系统内某些智能体的值相同,那么他们在这个模型中当成一个独立

的智能体

2. 这种模型是公式(1)的特殊情况

3. 接下来我们考虑一种平衡稳定系统:

(1) 设 x 数组包含平衡状态下所有聚集点

(2) 引入一个扰乱智能体,他的权重无限小

(3) 让系统继续演变,直至平衡

(4) 定义新旧系统间的距离

(5) 通过数学推导证明平衡稳定系统的条件

分析可知

A. 平衡稳定系统有两种情况:AB 集群的权重相等且他们之间的距离大于等于 2,或者 AB 集群的权重不相等切他们之间的距离严格大于一个特定值

B. 如果在不考虑扰乱智能体的微小权重的情况下,新旧系统间的距离趋近无穷小,那么我们称这个系统是平衡稳定系统

C. 反之,如果一个平衡系统会因为一个权重无穷小的扰乱智能体与新平衡系统有大于无穷小的距离,那么这个平衡系统就不是稳定的

D. 如果有稳定确定的初始值分布,那么平衡后的系统是平衡稳定系统

补充

I. 分析中提到的特定值,以及特定值的推导如下:

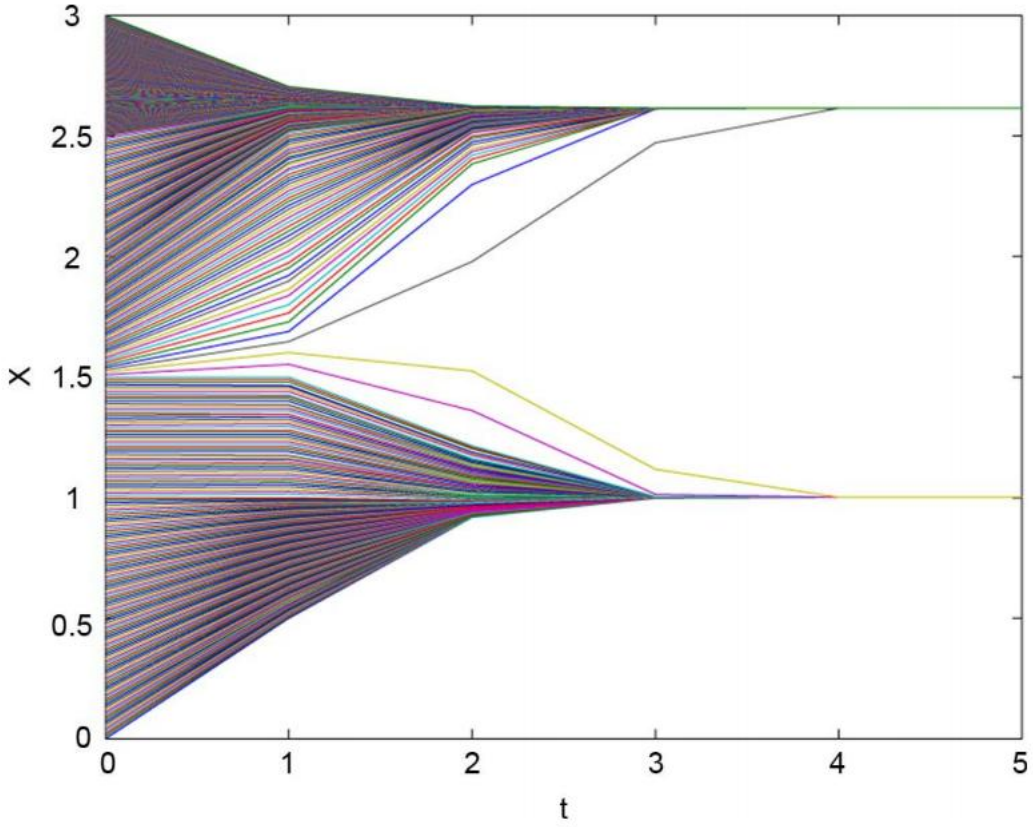
$$|x_A - x_B| > 1 + \frac{\min(W_A, W_B)}{\max(W_A, W_B)}$$
$$\text{if and only if}$$
$$\max\{|m - x_A|, |m - x_B|\} > 1.$$

$$m = (W_A x_A + W_B x_B)/(W_A + W_B)$$

假设 xa>xb,我们可以利用放缩的方法得到上式

平衡稳定系统实验

1. 设有 251 个智能体均匀分布在[0.2.5]上,500 个智能体均匀分布在[2.5,3]上,以世代 t 为横轴,各个智能体的位置为纵轴,画出他们迭代的过程图,如图:



2. 此时两个集群之间没有智能体,使系统维持平衡稳定状态

3. 因为两个区间不是一致的均匀分布,两个集群的权重并不相等

4. 集群间的距离满足分析时的关系:当两个集群的权重不相等时,两个集群的距离 1.6138 > 1.2559 = 1 + 153/598

5. 猜想一个拥有连续随机且独立的初始值多智能体系统,它收敛到稳定系统的概率接近 1,这一猜想可在论文中的连续模型中得证

实验复现

A. Fig. 2

$$x_i(t+1) = \frac{\sum_{j:|x_i(t)-x_j(t)|<1} x_j(t)}{\sum_{j:|x_i(t)-x_j(t)|<1} 1}. \qquad (1)$$

1. 依据公式(1),使用 python 源生的循环和 list。

(1) 先用 range 创造了一个均匀分布在[0,25]上的初始值,再用 0 填充了一个相同规格的列表。

(2) 用 while true 进入一个无限循环

(3) 用一个外面的 for 循环遍历列表中的每一个数,用于更新他们的值。

(4) 在循环里面还有一个 for 循环,用于寻找差值为 1 的元素,将他们累加到 sum 变量上,并记录满足元素的数量

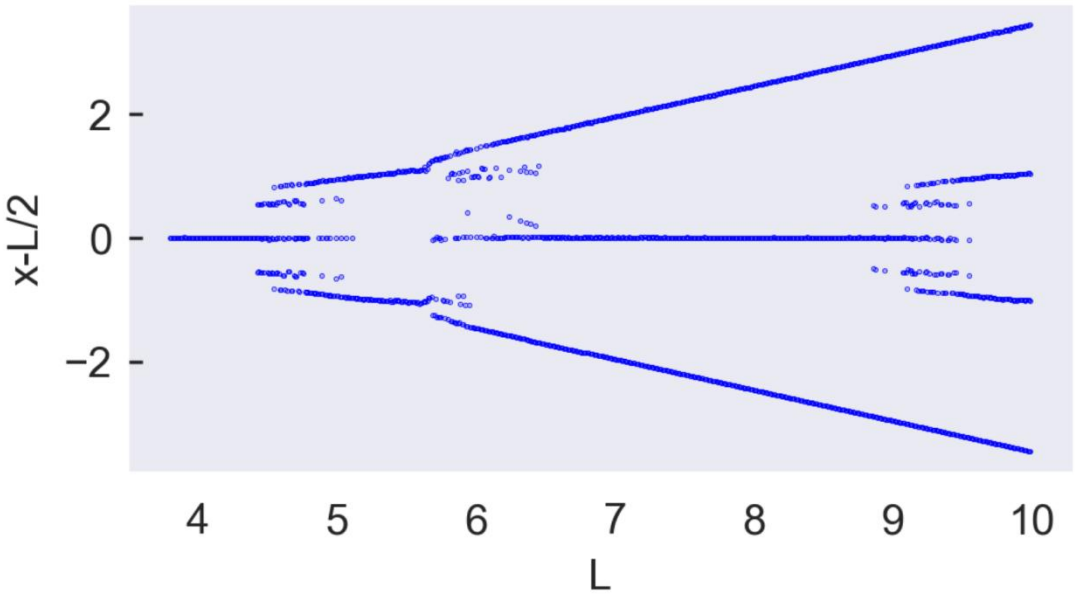
(5) 在里面的 for 循环遍历完之后,计算平均值,赋值给新数组,让外面的 for 循环向后迭代

(6) 当外面的 for 循环迭代完后,再遍历一次原数组的每个元素,检查是否和更新数组的每个元素相等

(7) 若相等即表示系统趋于平衡,可以跳出循环;若有元素不相等,那么将更新数组的值赋给原数组后重复上述循环直至满足条件

(8) 在画图方面,选择先去重再画图,效率会提升

(9) 但跑了一次之后发现效率极低,思考优化算法



2. 使用 numpy 库进行优化

(1) 用 np.arange 替代 range 创建的 list,并直接让用于更新的列表.copy 初始列表:

```
Step_Length = 2e-4
current_data = np.arange(0, L + Step_Length, Step_Length)
next_data = current_data.copy()
```

(2) 同 1(2)

(3) 同 1(3)

(4) 直接利用 numpy 的广播机制,用原数组减去当前循环的值,得到一个差值数组

(5) 用 begin 和 end 两个变量从 0 开始遍历差值数组,找从 0 开始第一个小于 1 的索引和从第一个小于 1 的索引开始,第一个大于 1 的索引

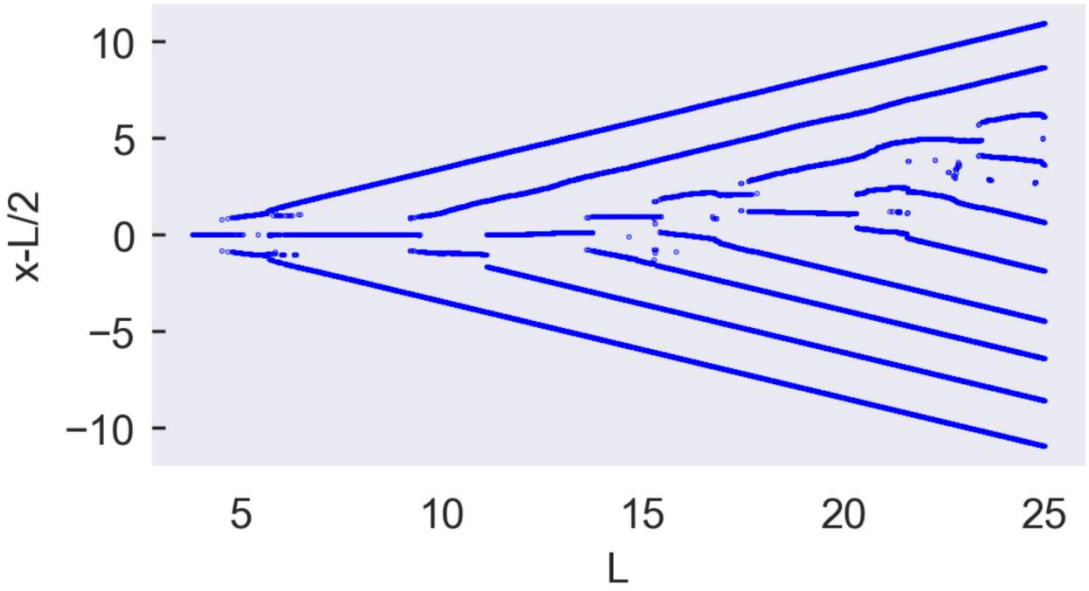
```
begin = 0
while diff[begin] >= 1 and begin < len(current_data):
    begin += 1

end = begin
while end < len(current_data) and diff[end] < 1:
    end += 1
```

(6) 利用数组的切片和.sum()方法,求这段序列的和,并用 end 和 begin 确定长度,得到平均值用于更新数组的值,然后让 for 循环向后迭代

(7) 不用遍历检查元素,而是使用 numpy 的 equal 方法检查一致性

```
if  np.array_equal(next_data, current_data):
    break
```

(8) 优化后发现效率大大提升,但仍然不够快,但是尾端存在少许偏移,思考从数学公式本身来优化



3. 借助图像,程序 debug 深入理解数学公式

(1) 智能体的值是逐渐聚集的,也就是说有些智能体聚集的早,有些智能体聚集的晚。而早聚集的智能体在之后的迭代中,行为是保持一致的,即聚集之后就不会再分开

(2) 因此可以做出这项优化:当外部循环中上一次循环智能体的原值与这次循环智能体的原值相等时,上一次循环智能体的更新值就可以直接赋给这次循环智能体的更新值:

```
if agent != 0 and current_num == current_data[agent-1]:
    next_data[agent] = next_data[agent-1]
    continue
```

(3) 在论文中可知,索引大的智能体的值一定大于等于索引小的智能体的值,因此大索引的智能体满足差值条件的范围一定比小索引智能体的范围往后一点。在算法中体现如下(last_begin 的值初始化为 0):

```
begin = last_begin
while diff[begin] >= 1 and begin < agent:
    begin += 1
last_begin = begin
```

(4) 在 end 值的搜寻上,因为智能体自身肯定与自身的距离小于 1,所以 end 不必从 begin 开始遍历,而是从自身开始遍历:

```
end = agent + 1
while end < len(current_data) and diff[end] < 1:
    end += 1
```

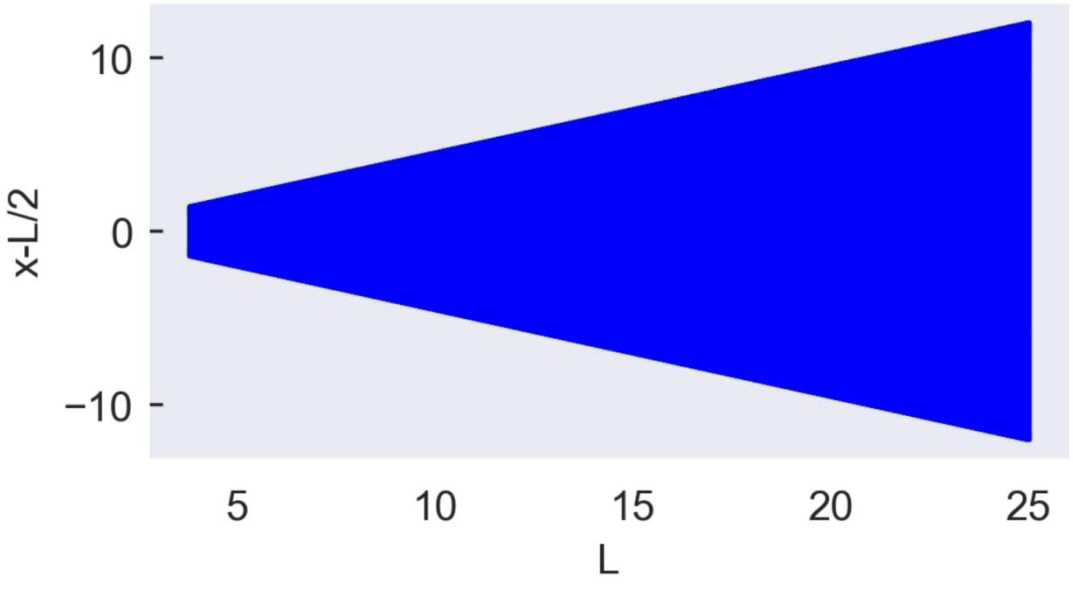
(5) 以上所有优化经过测试均可行

(6) 在程序 debug 中我了解到,数组的值是从两边向中间变化的,而且变化的区间应是对称的。在第一次迭代中,初始值大于 1 小于 n-1 的智能体的值不会变化,在第二次迭代中初始值大于 2 小于 n-2 的智能体的值不会变化,以此类推

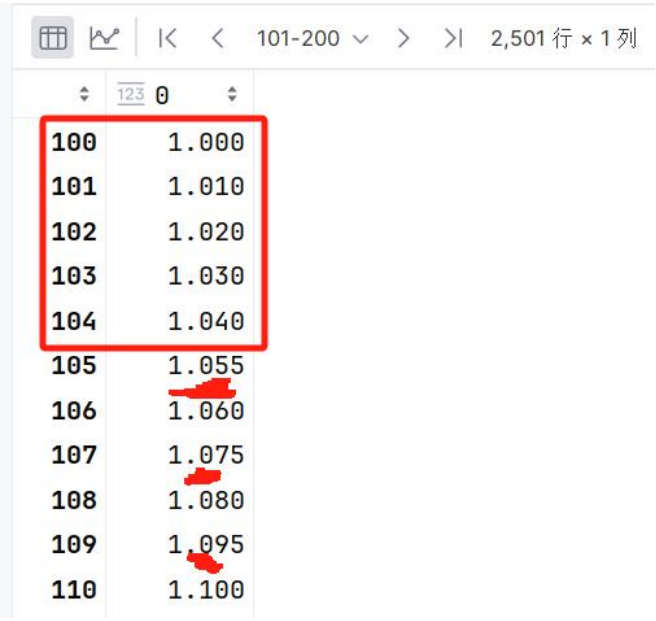
(7) 所以在算法上,可以每次检测原值和更新值,如果原值等于更新值,那么就可以将中间的值全部复制给新值,然后直接跳到尾部变化区间,更新尾部的值:

```
if skip == 0 and agent != 0 and current_data[agent-1] == next_data[agent-1]:
    next_data[agent:len(current_data) - agent] = current_data[agent:len(current_data) - agent].copy()
    agent = len(current_data) - agent
    skip = 1
    continue
```

(8) 但结果却是这样的:



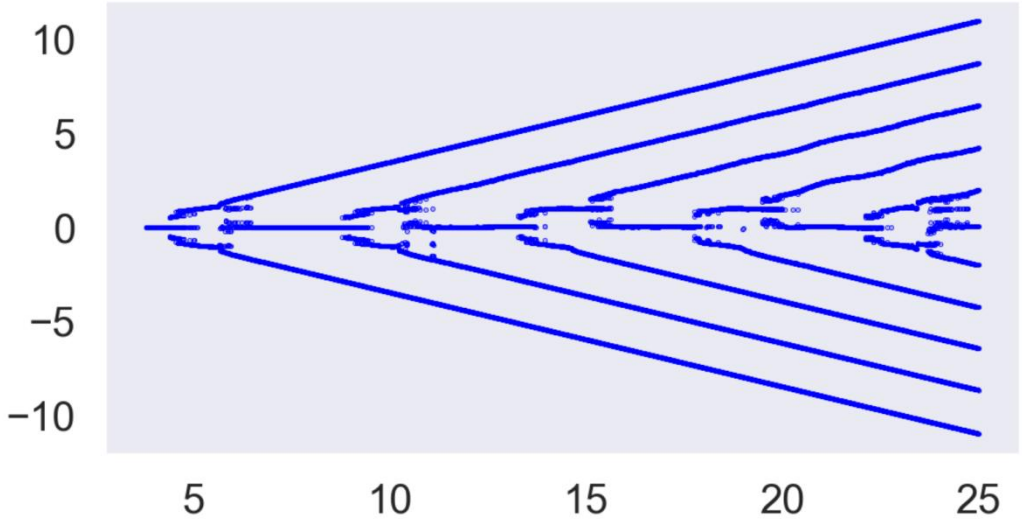
(9) 再 debug，我发现有些值在它不该变的循环中进行了微小的变化，如小数点最后一位变大或变小，如图（第一次迭代后的结果）：



(10) 排查原程序，猜测应是每次除法带来的精度误差，用 debug 验证，确实如此。因此，这种优化角度在这里不再适用，由于精度误差，每次的变化区间变得不再对称，也就无法根据前半部分的变化区间推出后半部分的变化区间

```
s = current_data[begin:end].sum()
next_data[agent] = round(s / (end - begin),2)
```

(11) 对于图像里尾部有偏移的现象，猜测也是精度误差带来的结果，所以每次赋新值时，只取两位小数，再画图：



(12)　偏移的情况好了一些,这就是我的最终结果了

B. Fig.3 和 Fig.4

$$x_i(t+1) = \frac{\sum_{j:|x_i(t)-x_j(t)|<1} x_j(t)}{\sum_{j:|x_i(t)-x_j(t)|<1} 1}. \qquad (1)$$

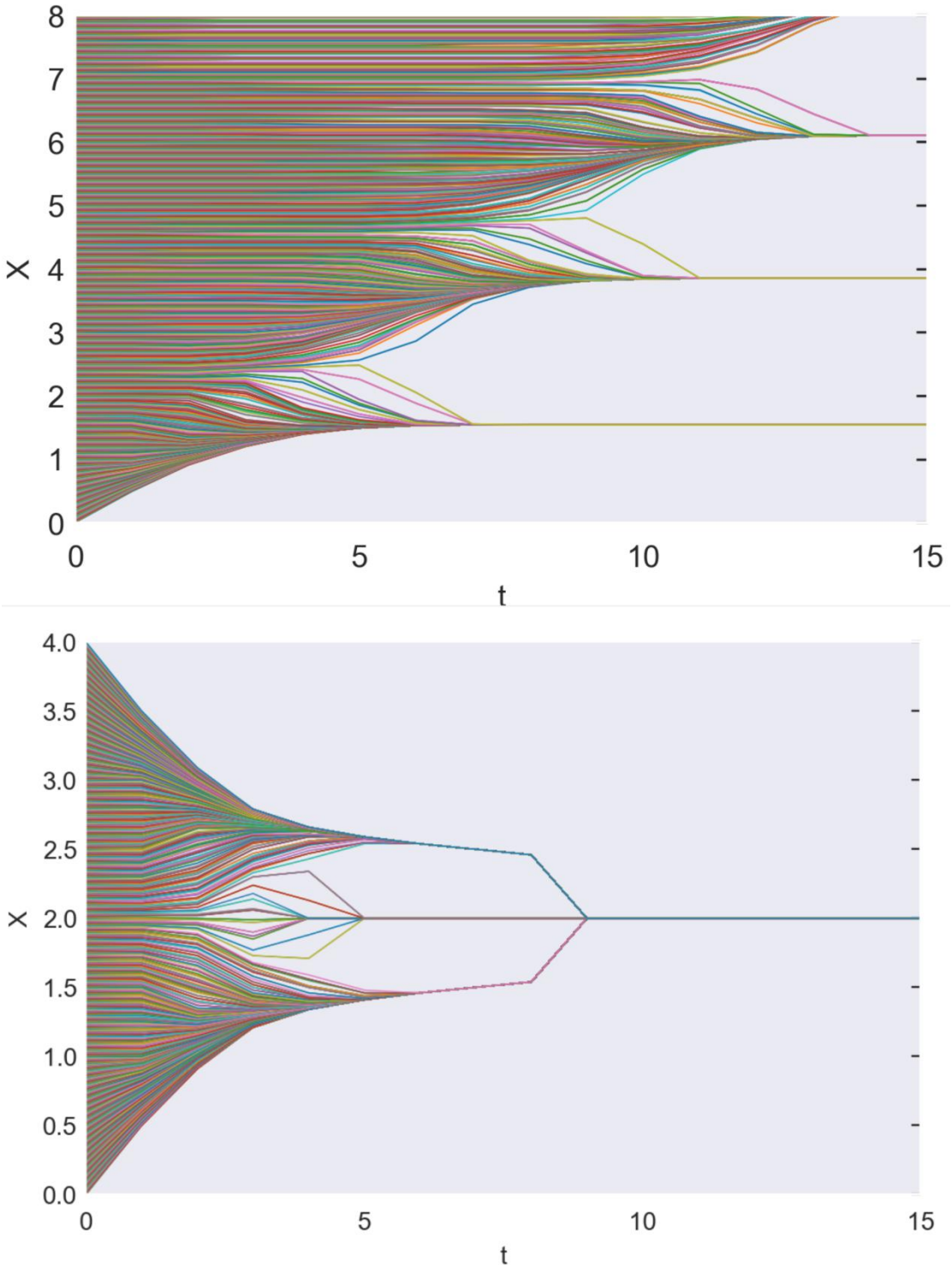
1. 仍然用到公式(1),大部分都可以沿用Fig.2的代码,区别是Fig.2是(0,L)的多组实验数据,而Fig.3和Fig.4是针对单一L的一组实验数据,需要在初始值上做些改变,如将数组变为二维数组记录每一次迭代的数据

```
def scope(start,L,Step_Length,fre):
    origin_data = np.arange(start, L + Step_Length, Step_Length)

    point = int((L - start)/ Step_Length) + 1
    ret_data = np.zeros((point,fre + 1))

    ret_data[:,0] = origin_data

    return origin_data, ret_data
```

2. 并将 while true 的无限循环改成,实验要求的有限次循环

```
def g(origin_data, ret_data,fre):
    count = 0
    while count < fre:
```

3. 就可以轻松得到两个实验结果:



C. Fig. 5

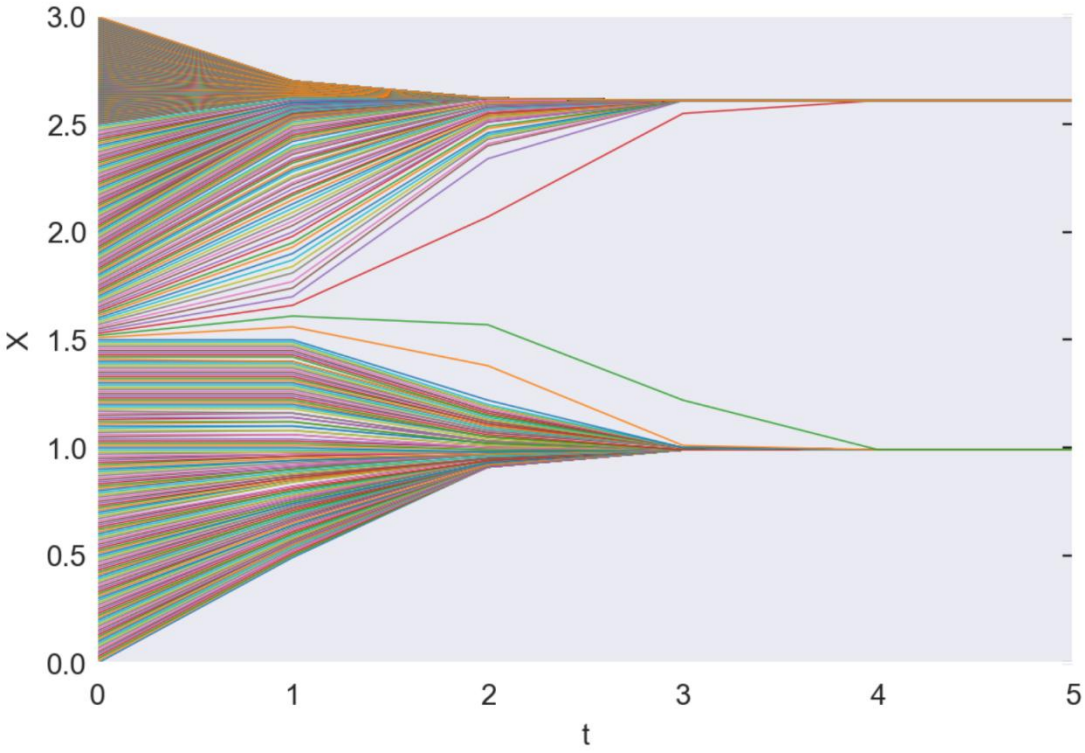
1. 图像 5 的实验是一个探究平衡稳定系统的实验，他与 Fig. 3 和 4 的区别在于，他的权重不等，[0, 2.5]的权重较小而[2.5, 3]的权重较大。

2. 权重不等体现在初始值的分布上，权重小的分布间隔为 0.01，而权重大的

分布间隔为 0.001

Fig. 5. Example of convergence to a stable equilibrium where the clusters are separated by less than 2. The initial distribution of opinions is obtained by taking 251 uniformly spaced opinions on $[0, 2.5]$ and 500 uniformly opinions on $[2.5, 3]$. Opinions converge to two clusters with 153 and 598 agents, respectively, that are separated by a distance $1.6138 > 1.2559 = 1 + 153/598$. Similar results are obtained when larger number of agents are used, provided that the initial opinions are distributed in the same way, i.e, with a density on $[2.5, 3]$ which is ten times larger than the density on $[0, 2.5]$.

3. 在程序方面,只需要把初始数组分成两次初始化,一次初始化[0,2.5],一次初始化[2.5,3],将他们连接后再传给迭代函数即可:



总结

1. 通过这个多智能体系统首先可以得到”物以类聚”这条成语,当某些智能体交流多次后他们就会聚集

2. 再来思考对实际生产的意义,实验中的位置值可以根据实际情况更改成其他值。他们最后聚集的结果提示了我们在现实中智能体能够通过交流改变自己的某个值,并在最后趋于一致