元胞自动机主要两种用途

https://blog.csdn.net/weixin_43102634/article/details/102996254

类森林火灾型

森林火灾的元胞自动机模型有三种状态：空位，燃烧着的树木及树木。则某元胞下一时刻状态由该时刻本身的状态和周围四个邻居的状态以一定的规则确定，规则如下：

1）如果某树木元胞的4个邻居有燃烧着的，那么该元胞下一时刻的状态是燃烧着的。

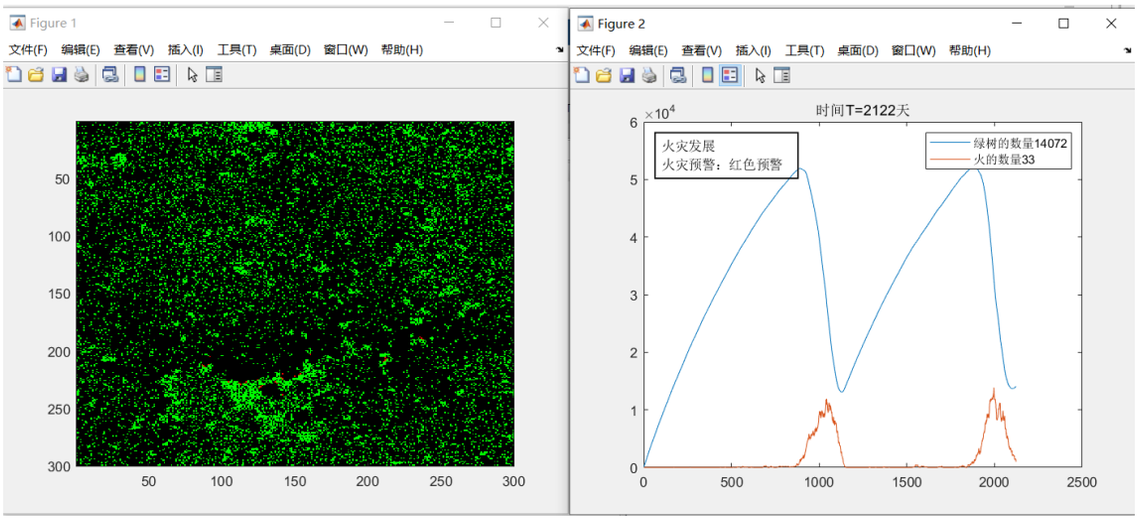
2）一个燃烧着的元胞在下一时刻变成空位。

3）所有树木元胞以一个低概率开始燃烧（模拟闪电引起的火灾）

4）所有空元胞以一个低概率变成树木（以模拟新的树木的生长）

```
close;
clear;
clc;
n = 300;    %元胞矩阵大小
Plight = 0.000001; Pgrowth = 0.001;
UL = [n 1:n-1];
DR = [2:n 1];
veg = zeros(n,n);       %初始化
% The value of veg:
% empty == 0
% burning == 1
% green == 2
imh = image(cat(3,veg,veg,veg));
m=annotation('textbox',[0.1,0.1,0.1,0.1],'LineStyle','-','LineWidth',1,'String','123');
for i = 1:100000
   sum = (veg(UL,:) == 1) + (veg(:,UL) == 1) + (veg(DR,:) == 1) + (veg(:,DR) == 1);
   %根据规则更新森林矩阵：树 = 树 - 着火的树 + 新生的树
   veg = 2 * (veg == 2) - ( (veg == 2) & (sum > 0 | (rand(n,n) < Plight)) ) + 2 * ( (veg == 0) &
rand(n,n) < Pgrowth);
   a=find(veg==2);
   b=find(veg==1);
   aa=length(a);
   bb=length(b);
   shu(i)=aa;
   fire(i)=bb*30;
   if (bb>=0&&bb<=10)
      str1='森林正常';
   elseif (bb>10&&bb<=100)
      str1='火灾发展';
   elseif (bb>100)
      str1='森林大火';
   end
   if ((aa>48000)||(bb>=10))
     str2='火灾预警：红色预警';
```

```
    elseif (aa>42000&&aa<=48000)
        str2='火灾预警：黄色预警';
    elseif (aa>35000&&aa<=42000)
        str2='火灾预警：蓝色预警';
    elseif (aa>=0&&aa<=35000)
        str2='火灾预警：安全';
    end
    str=[str1 10 str2];
    set(imh, 'cdata', cat(3, (veg == 1), (veg == 2), zeros(n)) )
    drawnow
    figure(2)
    delete(m)
    plot(shu);
    hold on
    plot(fire);
    legend(['绿树的数量',num2str(aa)],['火的数量',num2str(bb)]);
    title(['时间T=',num2str(i),'天']);
    m=annotation('textbox',[0.15,0.8,0.1,0.1],'LineStyle','-','LineWidth',1,'String',str);
    hold off
%     pause(0.0001)
end
```



单方向车流流量模拟

```
clc
clear;
%build the GUI
%define the plot button
plotbutton=uicontrol('style','pushbutton','string','Run',
'fontsize',12,'position',[100,400,50,20],'callback','run=1;');
%define the stop button
```

```
erasebutton=uicontrol('style','pushbutton','string','Stop','fontsize',12,'position',[200,400,50,
20],'callback','freeze=1;');
%define the Quit button
quitbutton=uicontrol('style','pushbutton','string','Quit','fontsize',12,'position',[300,400,50,20
],'callback','stop=1;close;');
number=uicontrol('style','text','string','1','fontsize',12,'position',[20,400,50,20]);
%CAsetup
n=1000; %数据初始化
z=zeros(1,n); %元胞个数
z=roadstart(z,200); %道路状态初始化，路段上随机分布200辆
cells=z;
vmax=5; %最大速度
v=speedstart(cells,vmax); %速度初始化x=1; %记录速度和车辆位置
x=1;
memor_cells=zeros(3600,n);
memor_v=zeros(3600,n);
imh=imshow(cells); %初始化图像白色有车，黑色空元胞
set(imh,'erasemode','none')
axis equal
axis tight
stop=0; %wait for a quit button push
run=0; %wait for a draw
freeze=0; %wait for a freeze (冻结)
while (stop==0 && x<1102)
if(run==1)
    %边界条件处理，搜素首末车，控制进出，使用开口条件
    a=searchleadcar(cells);
    b=searchlastcar(cells);
%     [cells,v]=border_control(cells,a,b,v,vmax);
    i=searchleadcar(cells); %搜索首车位置
    for j=1:i
    if (i-j+1==n)
    [z,v]=leadcarupdate(z,v);
        continue;
      else
    %==========================加速、减速、随机慢化
    if cells(i-j+1)==0 %判断当前位置是否非空
       continue;
    else
      v(i-j+1)=min(v(i-j+1)+1,vmax); %加速
    %======================减速
    k=searchfrontcar((i-j+1),cells); %搜素前方首个非空元胞位置
    if(k==0) %确定与前车之间的元胞数
    d=n-(i-j+1);
```

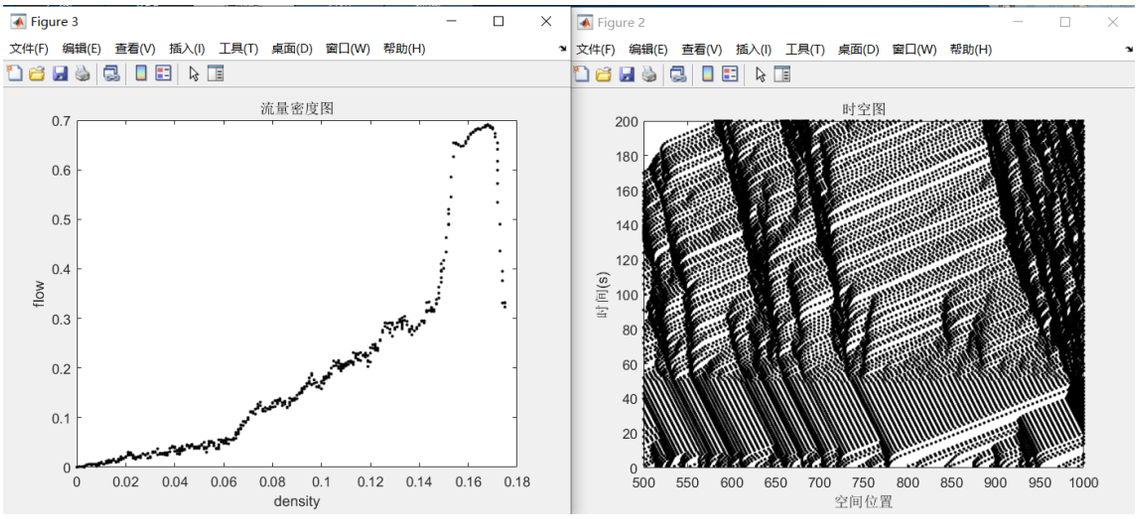
```
    else
      d=k-(i-j+1)-1;
    end
    v(i-j+1)=min(v(i-j+1),d);%减速
    %随机慢化
    v(i-j+1)=randslow(v(i-j+1));
    new_v=v(i-j+1);
%更新车辆位置
    z(i-j+1)=0;
    z(i-j+1+new_v)=1;
    %更新速度
    v(i-j+1)=0;
    v(i-j+1+new_v)=new_v;
    end
    end
    end
    cells=z;
    memor_cells(x,:)=cells; %记录速度和车辆位置
    memor_v(x,:)=v;
    x=x+1;
    set(imh,'cdata',cells) %更新图像
    %update the step number diaplay
pause(0.0001);
stepnumber=1+str2num(get(number,'string'));
set(number,'string',num2str(stepnumber))
end
if (freeze==1)
run=0;
freeze=0;
end
drawnow
end
figure(2)
for l=1:1:200
for k=500:1:1000
if memor_cells(l,k)>0
  plot(k,l,'k.');
hold on;
end
end
end
xlabel('空间位置');
ylabel('时间(s)');
title('时空图');
```

```
    for i=1:1:500
    density(i)=sum(memor_cells(i,:)>0)/1000;
    flow(i)=sum(memor_v(i,:))/1000;
    end
    figure(3)
    plot(density,flow,'k.');
    title('流量密度图')
    xlabel('density')
    ylabel('flow')

 % 函数: speedstart.m程序代码
function [v_matixcells]=speedstart(matrix_cells,vmax)
  %道路初始状态车辆速度初始化
  v_matixcells=zeros(1,length(matrix_cells));
  for i=1:length(matrix_cells)
    if matrix_cells(i)~=0
    v_matixcells(i)=round(vmax* rand(1));
    end
  end
end

%函数: searchleadcar.m程序代码
function [location_leadcar]=searchleadcar(matrix_cells)
  i=length(matrix_cells);
  for j=1:i
    if matrix_cells(i-j+1)~=0
      location_leadcar=i-j+1;
      break;
    else
      location_leadcar=0;
    end
  end
end

%函数: leadcarupdate.m程序代码
function [new_matrix_cells,new_v]=leadcarupdate(matrix_cells,v)
%第一辆车更新规则
  n=length(matrix_cells);
  if v(n)~=0
    matrix_cells(n)=0;
    v(n)=0;
  end
  new_matrix_cells=matrix_cells;
  new_v=v;
```

```
end

%函数: randslow.m程序代码
function[new_v]=randslow(v)
  p=0.3;%慢化概率
  rand('state',sum(100*clock)*rand(1));
  p_rand=rand; %产生随机概率
  if p_rand<=p
    v=max(v-1,0);
  end
  new_v=v;
end


%函数: roadstart.m 程序代码
function [matrix_cells_start]=roadstart(matrix_cells,n)
%道路上的车辆初始化状态，元胞矩阵随机为0或1, matrix_cells 初始矩阵，n初始车辆数
  k=length(matrix_cells);
  z=round(k*rand(1,n));
  for i=1:n
    j=z(i);
    if j==0
      matrix_cells(j)=0;
    else
      matrix_cells(j)=1;
    end
  end
  matrix_cells_start=matrix_cells;
end

% 函数：searchfrontcar.m 程序代码
function [location_frontcar]=searchfrontcar(current_location,matrix_cells)
  i=length(matrix_cells);
  if current_location==i
    location_frontcar=0;
  else
    for j=current_location+1:i
      if matrix_cells(j)~=0
      location_frontcar=j;
      break;
      else
      location_frontcar=0;
      end
     end
```

```
    end
end
```

```
%函数: searchlastcar.m程序代码
function [location_lastcar]=searchlastcar(matrix_cells)
%搜索尾车位置
for i=1:length(matrix_cells)
    if matrix_cells(i)~=0
        location_lastcar=i;
        break;
     else %如果路上无车，则空元胞数设定为道路长度
         location_lastcar=length(matrix_cells );
    end
end
end
```



F:\美赛\03模型算法大全\03模型算法大全（30+种常用算法模型+课件讲义代码）\元胞自动机\车流模拟

14年美赛A题

问题A：除非超车否则靠右行驶的交通规则

在一些汽车靠右行驶的国家（比如美国，中国等等），多车道的高速公路常常遵循以下原则：司机必须在最右侧驾驶，除非他们正在超车，超车时必须先移到左侧车道在超车后再返回。

建立数学模型来分析这条规则在低负荷和高负荷状态下的交通路况的表现。你不妨考察一下流量和安全的权衡问题,车速过高过低的限制，或者这个问题陈述中可能出现的其他因素。这条规则在提升车流量的方面是否有效？如果不是，提出能够提升车流量、安全系数或其他因素的替代品（包括完全没有这种规律）并加以分析。

在一些国家，汽车靠左形式是常态，探讨你的解决方案是否稍作修改即可适用，或者需要一些额外的需要。

最后，以上规则依赖于人的判断，如果相同规则的交通运输完全在智能系统的控制下，无论是部分网络还是嵌入使用的车辆的设计，在何种程度上会修改你前面的结果？