2014美赛相关MATLAB程序（基于NS模型）

主程序：NaSch\_3.m程序代码

% 单车道 最大速度3个元胞 开口边界条件 加速 减速 随机慢化

clf

clear all

%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]);

%CA setup

n=100; %数据初始化

z=zeros(1,n); %元胞个数

z=roadstart(z,5); %道路状态初始化，路段上随机分布5辆

cells=z;

vmax=3; %最大速度

v=speedstart(cells,vmax); %速度初始化

x=1; %记录速度和车辆位置

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)

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; %确定于前车之间的元胞数

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);

%======================================加速、减速、随机慢化

%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(0.2);

stepnumber = 1 + str2num(get(number,'string'));

set(number,'string',num2str(stepnumber))

end

if (freeze==1)

run = 0;

freeze = 0;

end

drawnow

end

///////////////////////////////////////////////////////////////////////

函数：border\_control.m程序代码

Function[new\_matrix\_cells,new\_v]=border\_control(matrix\_cells,a,b,v,vmax)

%边界条件，开口边界，控制车辆出入

%出口边界，若头车在道路边界，则以一定该路0.9离去

n=length(matrix\_cells);

if a==n

rand('state',sum(100\*clock)\*rand(1));%¶¨ÒåËæ»úÖÖ×Ó

p\_1=rand(1); %产生随机概率

if p\_1<=1 %如果随机概率小于0.9，则车辆离开路段，否则不离口

matrix\_cells(n)=0;

v(n)=0;

end

end

%入口边界，泊松分布到达，1s内平均到达车辆数为q，t为1s

if b>vmax

t=1;

q=0.25;

x=1;

p=(q\*t)^x\*exp(-q\*t)/prod(x); %1s内有1辆车到达的概率

rand('state',sum(100\*clock)\*rand(1));

p\_2=rand(1);

if p\_2<=p

m=min(b-vmax,vmax);

matrix\_cells(m)=1;

v(m)=m;

end

end

new\_matrix\_cells=matrix\_cells;

new\_v=v;

///////////////////////////////////////////////////////////////////////

函数：leadcarrupdate.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;

///////////////////////////////////////////////////////////////////////

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));%¶¨ÒåËæ»úÖÖ×Ó

p\_rand=rand; %产生随机概率

if p\_rand<=p

v=max(v-1,0);

end

new\_v=v;

///////////////////////////////////////////////////////////////////////

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;

///////////////////////////////////////////////////////////////////////

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

///////////////////////////////////////////////////////////////////////

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

///////////////////////////////////////////////////////////////////////

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

///////////////////////////////////////////////////////////////////////

函数：speadstart.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